REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 10/19/01	2. REPORT TYPE FINAL	3. DATES COVERED (From - To) 6/15/99-9/14/01
4. TITLE AND SUBTITLE Object-oriented PIC code with Upgraded physics and platform- Independent GUI		5a. CONTRACT NUMBER F49620-99-C-0028 5b. GRANT NUMBER FQ8671-99012717 5c. PROGRAM ELEMENT NUMBER N/A
6. AUTHOR(S) Dr. David Bruhwiler		5d. PROJECT NUMBER N/A 5e. TASK NUMBER N/A 5f. WORK UNIT NUMBER N/A
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Tech-X Corporation 5541 Central Ave #135 Boulder CO 80301 303-448-0727 University of California – Berkeley University of Michigan		8. PERFORMING ORGANIZATION REPORT NUMBER 7010
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office for Scientific Research		10. SPONSOR/MONITOR'S ACRONYM(S) ASOSR 11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution Statement A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

20011123 012

14. ABSTRACT

This report covers the period between June 15, 1999 and September 14, 2001. We have developed a new graphical user interface (GUI) for the particle-in-cell (PIC) code OOPIC, providing a powerful new cross-platform simulation tool for designers of high-power microwave (HPM) devices. Algorithms for modeling the following physical effects have been developed and implemented within the new version of OOPIC: the electron self-force; secondary emission of electrons from metal surfaces; improved particle loading and injection; improved charge and current deposition for curvilinear grid coordinates; and digital filtering for noise reduction in cylindrical coordinates. The accuracy of the basic PIC algorithm has been analyzed for electrostatic and electromagnetic field solves, showing that the combined errors for each time step are second-order in both time and space, when taking the limit of many macro-particles. Previous work on fluctuation reduction algorithms for direct-simulation Monte Carlo (DSMC) codes has been applied for the first time to a 1-D electrostatic PIC code, showing that this technique holds promise for significant noise reduction in PIC simulations.

15. SUBJECT TERMS
Key words: high-power microwave; particle-in-cell; graphical user interface

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	14	John R Cary 19b. TELEPHONE NUMBER (include area code) 303-448-0727

Tech-X Corporation

October 19, 2001

Final Report

STTR Phase II project, contract number F49620-99-C-0028
"Object-oriented PIC code with upgraded physics and platform-independent GUI"
David Bruhwiler, Principal Investigator; (303) 448-0732, bruhwile@txcorp.com

Objectives:

The objectives of this project have not changed since the last status report.

Status of Effort:

This Phase II STTR project was completed on schedule on September 14, 2001.

Accomplishments / New Findings:

This report covers the period between June 15, 1999 and September 14, 2001. Detailed technical information on the research described is not presented here, but can be found in attached reports and journal publications and in portions of the recent text edited by R.J. Barker and E. Schamiloglu [1]. Many of the algorithms discussed here have been implemented in the XOOPIC code [2], which is freely available for noncommercial use – thus, providing benefit to the entire PIC community.

The Electron Self-Force

In this section, work on the electron self-force issue is summarized. The attached report, "A Radiation Damping Algorithm for Particle Simulation", describes this work in more detail. This report will be submitted for publication by the authors.

The electron self-force cannot be modeled by a conventional PIC algorithm, due to the finite resolution of temporal and spatial scales. A sophisticated analytical treatment of self-force effects on charged particles has been developed previously [3]. However, a recent report [4] has applied this work to the problem of an electron beam in an HPM tube, showing that self-force effects are negligible in microwave HPM devices.

The full equations from Ref. [3] describing the self-force are quite complicated and would be computationally expensive to include in a PIC algorithm. More seriously, these equations include the second time derivative of the particle momentum -- a very noisy quantity in a time-explicit electromagnetic PIC algorithm. Thus, it is not practical to directly implement the results of Ref. [3] in a PIC code.

In the physical limit where self-force effects are important (i.e., for ultra-relativistic electrons), one can approximate the self-force as the classical recoil of the particle upon emitting synchrotron radiation. The equations describing this effect [5] are relatively simple and require only the first time derivative of the particle momentum. These equations have been successfully implemented and tested in XOOPIC. It is unlikely that this work will improve PIC simulations of HPM tubes in the near term; however, it is relevant to beam devices in which ultra-relativistic electrons interact with strong magnetic fields.

We also explored the issue of synchrotron radiation effects (i.e. bremsstrahlung) during collisions between beam electrons and a tenuous background neutral gas. An unpublished report [6] directly addresses this issue for many monatomic gasses. This physics could be accurately modeled in XOOPIC through a Monte Carlo collision mechanism, which would be more appropriate than any direct modification to the electromagnetic PIC algorithm. However, Ref. [6] shows that energy losses due to bremsstrahlung are negligible until the electron energy exceeds 100 MeV and the neutral gas density becomes much larger than is found in an evacuated tube. Thus, we do not plan to further pursue this issue.

Secondary Emission of Electrons

In this section, the work on the secondary emission model is summarized. The complete details of the model can be found in Sec. 11.4.2, beginning on p. 397 of Ref. [1].

Secondary emission can play an important role in many devices where electrons, ions, or neutral particles impact surfaces. The impact can result in reflection of the incident particle, multiple scattering events within the surface lattice followed by ejection, or transfer of the incident energy to electrons in the lattice leading to ejection of one or more electrons.

The secondary model, implemented in the XOOPIC code, includes a simple model and a more complete model. In the simple model, an energy and angular-independent secondary emission coefficient is specified. The emitted electrons are modeled as true secondaries, ejected with a uniform angular distribution at a specified temperature. The complete model allows specification of the key parameters to model energy and angular dependence, relevant to most conductive materials and many dielectric materials as well. The complete model also considers surface roughness in computing the secondary emission coefficient. In the complete model, the fractions of reflected primaries, scattered primaries, and true secondaries can be specified, along with the energy and angular distribution functions for the true secondaries. The current model defaults to isotropic angular distribution, which is appropriate for most metals. Fractional yields are emitted statistically using pseudorandom numbers.

Secondary emission plays an important role in many devices relevant to microwave sources, including two-surface RF multipactor phenomena, single surface DC multipactor phenomena (such as window breakdown), as well as beam intercept current in collectors or circuit structures in microwave sources and accelerators, and ion/neutral-induced secondary emission in DC and RF discharges.

Loading and Injection Models

This section summarizes the work on loading and injection. A careful analysis of the methods for loading and injecting particles in PIC codes has indicated deficiencies in the accuracy of common methods. This work is described in detail in Ref. [7], where the analysis of existing methods is performed, and correction terms are proposed for second order accuracy for loading and injection in the general case.

A second-order accurate method for loading general phase space distributions of particles has been described. The method for time centering described in Birdsall and Landon [8] is shown to be first order accurate when the time-derivative of the acceleration is non-zero, the magnetic field is non-zero, or the initial velocity is non-zero. A correction is derived to make ensure second order accurate time centering of loaded particles for the general case.

The injection of particle flux from a boundary in the system was also considered. The injection may be due to a thermionic or field emitter, or it may be flux crossing a mathematical boundary line with a specified distribution outside the system. Analysis indicates that existing injection schemes result in a first order error term. In this work, second order accuracy was achieved for a number of special cases by adding additional terms to cancel the first order error. It was found that cancellation of error terms leads to an increasingly complicated algorithm as the

complexity of the terms and their derivatives in time and space increases. Analysis of the computational efficiency of the various correction schemes was also performed.

The loading and injection schemes are relevant to applications in which particles are loaded initially (such as discharges or plasma-filled microwave tubes), or injected from the boundaries of the system (such as the cathode in a microwave tube, or secondary emission from a collector). A number of corrective schemes are presented which minimize the increased complexity of the equations of motion for achieving second order accuracy.

Accuracy Analysis of the PIC Method

In this section, the analysis of the accuracy of the PIC model is summarized. More details are provided in three presentations [9,10,11], all of which are attached to this report.

The discretization errors for each component stage of the PIC method are derived for both electrostatic and electromagnetic models. Some of the error terms have been described previously; here the analysis is extended to include errors in the interpolation algorithm, which connect the continuum particles to the discrete fields. The terms coupling errors between PIC steps are also described. The error analysis includes both spatial and temporal errors.

For the electrostatic PIC model, the steps analyzed include the interpolation of position to a mesh to compute the charge density, the discretization error of the finite difference solution of Poisson's equation, errors in computing the gridded electric field from potential, errors in interpolating the electric field from the mesh to the continuum particle locations, and the error terms in integrating Lorentz's equation. Finally the individual errors are incorporated into a coupled error term. It was particularly interesting to note that the error terms in the linear charge weighting precisely cancel those in the Poisson equation. The total error for the linear weighting case was shown to contain error terms no worse than second order in both space and time, with dependence upon higher derivatives of the potential.

For the electromagnetic PIC model, the steps analyzed include the interpolation from the continuum positions and velocities of particles to the discrete locations of the Yee mesh to compute the current density, the errors in the finite difference discretization of Ampere's law, and Faraday's law, the errors in the interpolation from the Yee mesh location to common nodal locations, the errors from interpolating the nodal fields to the continuum particle positions, and the integration of the relativistic Lorentz equation.

Charge Weighting Correction

This section provides a summary of the work on charge weighting corrections for non-cartesian geometry. Complete details are provided in Ref. [12], which is attached to this report.

This novel formulation of the charge weighting algorithm for the particle-in-cell method was developed to eliminate a systematic error in the charge density in curvilinear systems which occurs for the classic PIC weighting method [8].

In the standard PIC scheme, particles exist in a continuum phase space, while the fields are defined on a discrete mesh. The particles and fields interact through various interpolation schemes. The source terms for the field equations are the current density, J, and the charge density, ρ. These quantities are computed weighting the current or charge for each particle to mesh nodes, and then dividing by the appropriate geometric terms to obtain the current or charge density. The geometric terms were viewed as a property of the mesh, independent of the interpolation scheme used for the particles. These schemes lead to systematic errors of 33% and 100% on axis in cylindrical and spherical coordinates, respectively, which remain even as the mesh size approaches the infinitesimal limit. Errors also occur at the outer radial edge, and at interior boundaries as well. The errors occur at every mesh point for non-uniform meshes. Each of these errors is shown in Figs. 1 and 2 in the attached preprint [12]. Case-dependent correction

factors have been described previously, but each of these is valid only for a specific mesh with a specific interpolation scheme.

In this work, we derive an improved weighting algorithm which requires no correction for the general case, valid for any mesh with any interpolation scheme. Rather than computing the volume or surface area elements from purely geometric considerations, this scheme employs an interpolation for computing the volume elements. The interpolation is symmetric with that used for weighting the charge/current to the mesh, and removes all systematic errors. The method has the desirable properties that total volume is conserved, as is total charge/current, and it yields the exact answer for the uniform charge distribution. Arbitrary distributions are correct to the mesh resolution, with error approaching zero as the mesh size approaches zero.

This method is relevant to any plasma device model in curvilinear coordinates, such as a cylindrical model of an O-type microwave tube. The new scheme will result in reduced noise and errors for particles near the axis and near any boundary as well, which means a less noisy RF output signal in the above case.

Digital Filter Noise Reduction

This section provides a summary of the work on digital filter noise reduction. Complete details are provided in Ref. [13], which is attached to this report..

Digital filters are used in many digital applications to reduce high frequency or short wavelength noise. The technique has been described for use in particle simulation for Cartesian coordinates [Birdsall85]. However, the method cannot be applied directly to particle simulation in cylindrical coordinates, a domain of great interest in microwave-beam devices, beam optics, plasma-filled devices, and many others. In these cylindrical systems, the noise on axis can be considerable when the number of simulation particles per cell is small; this is exacerbated by the cylindrical coordinate system where the number of particles per cell is proportional to the radius for a uniform density or current profile. In addition to the immediate effects of noise, the short wavelength noise associated with low numbers of computer particles per cell can lead to numerical heating.

We have extended the standard 1-2-1 digital filter to the cylindrical coordinate system, and the model has been implemented and tested in one dimension; it will be extended to two dimensions as part of an ongoing study. The filter operates on charge density in an electrostatic model, and on current density in an electromagnetic model.

The filter was designed to satisfy a number of desirable criteria. It conserves net charge and current exactly. It preserves a uniform distribution; i.e. charge/current does not migrate systematically towards the axis or outer radius. Furthermore, analysis indicates the filter has strength decreasing rapidly with radius, which can provide the benefits of filtering close to the axis with minimum impact of the sheath or beam profile at the outer radius. Due to the weakness of the filter, it was empirically observed that hundreds of passes were required for reduction of short wavelength noise that five to 10 passes of a planar 1-2-1 filter would provide. A pre-computed multi-pass filter matrix was devised which would apply an arbitrary number of passes with a single matrix multiply. The multi-pass filter was demonstrated to be far effective at reducing noise than the addition of particles.

The multi-pass digital filter described in the attached manuscript provides a number of benefits relevant to microwave-beam devices. It can be used to reduce particle-generated noise to acceptable levels, as well as reduction of noise and numerical-heating effects in plasmas in devices such as plasma-filled or focused microwave tubes, or in plasma discharges in cylindrical configurations.

Fluctuation Reduction Algorithms

This section provides a summary of the work on fluctuation reduction algorithms. Complete details are provided in the final report from U. Michigan [14] and the presentation by Prof. Boyd [15], both of which are attached to this report..

The objective of this work was to demonstrate the application of a direct simulation Monte Carlo (DSMC) [16] algorithm, called information preservation (IP), [17,18] to noise reduction in a 1-D particle-in-cell (PIC) code. This technique was successfully implemented in the 1-D electrostatic PIC code XES1, and effective noise reduction has been shown.

A new Graphical User Interface for XOOPIC

The initial version of the new cross-platform graphical user interface (GUI) for XOOPIC is complete. This professional GUI runs equally well under Microsoft Windows, Linux or any version of Unix. Work on the GUI is continuing under a project funded by DoE, and we plan to support the Macintosh OSX in the near future.

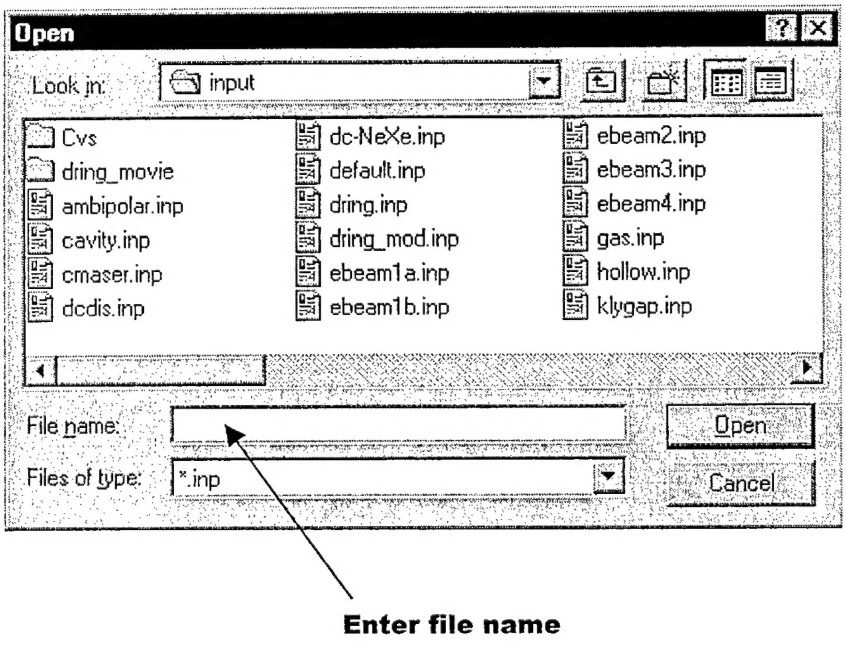
The GUI provides users with full control of the simulation and displays real-time 2-D and 3-D plots of relevant physical quantities. This interactive approach to PIC simulation provides insights that are difficult to obtain from the post-processing of data.

The commercial version of XOOPIC, to be called OOPIC Pro, is now ready for beta testing. A commercial licensing agreement has been negotiated with the University of California Berkeley, enabling Tech-X to satisfy the commercialization and technology transfer aspects of the STTR program.

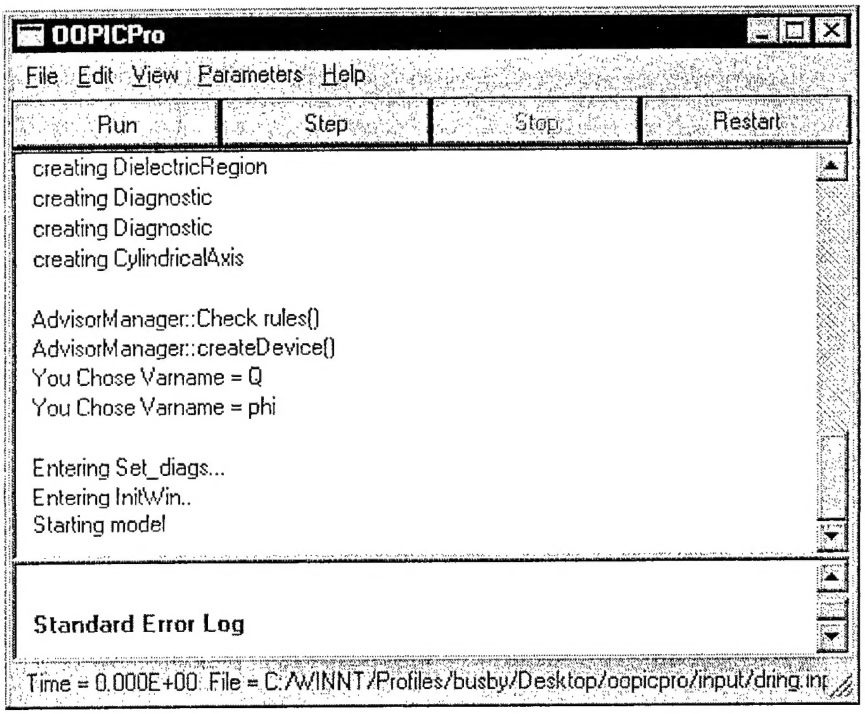
An initial user manual was developed for OOPIC Pro under this contract. Further work on the user manual is continuing under a project funded by DoE. An excerpt from the user manual is provided here, in order to show what has been accomplished.

Excerpt from the OOPIC Pro user manual:

Windows users begin using the code by double-clicking on the oopicpro.exe executable file. The following file dialog will appear:



Browse until you have found the directory containing the input file you wish to open. The defualt directory is oopicpro/input. Double click on the desired input file. Alternatively, click once on the file you want to open and click Open, or type the name of the file and hit <RETURN>. Upon choosing an input file, the main OOPIC Pro control window will appear:



At the top of the control window, there is a menu bar. Below the bar are four buttons which control the progress of the simulation. In the interior of the window are the two program logs. The current simulation time is given on the left side of the lower window border, and the name of the current input file on the right side of the border.

File menu

Clicking on the File menu gives the user access to Loading and Saving files. Although the user must specify an input file upon starting the program, one can Open another input file in order to begin a new simulation.

The File menu also contains an Exit heading. Clicking on Exit brings up a query window asking the user if he wants to end the program. One can also end the program immediately by clicking on the 'X' button in the upper right hand corner of the control window.

Edit menu

The Edit menu allows the user to Cut, Copy and Paste. The user can also view the input file in a read-only browser window by clicking on Input text file under the Edit window.

Under the View menu, choosing Style brings up a sub-menu with choices about the appearance of the windows in OOPIC Pro.

Also under the View->Logs submenu, one can choose whether to view the program logs.

After choosing the last heading under the View menu, Diagnostics, the list of available diagnostic plots is displayed. Clicking on any of these will bring up the corresponding diagnostic plot.

Parameter menu.

The first sub-menu under the parameter menu is denoted Iterations. Clicking on Maximum Number in that sub-menu brings up an value entry window, in which the user is able to set the maximum number of time steps in the simulation. The default value is '0', which specifies that there is no limit on the number of iterations. Change the value by typing another integer, or by clicking on the up or down arrows on the right edge of the value blank. When the entry blank contains the desired number, click the Ok button or hit <RETURN>.

Choosing Periodic dump brings up a similar value entry window in which the user can choose how often to periodically save the progress of the simulation. The default is '0', meaning no dump files are saved. More information about dump files later in this section of the manual.

Clicking on Number per update causes another value entry window to appear, in which the user can change the how many time steps are between graphics updates.

The second sub-menu in the Parameter menu is titled Movies. These parameters, as well as the entire process for creating a movie of the plots in a simulation, will be explained later in this section.

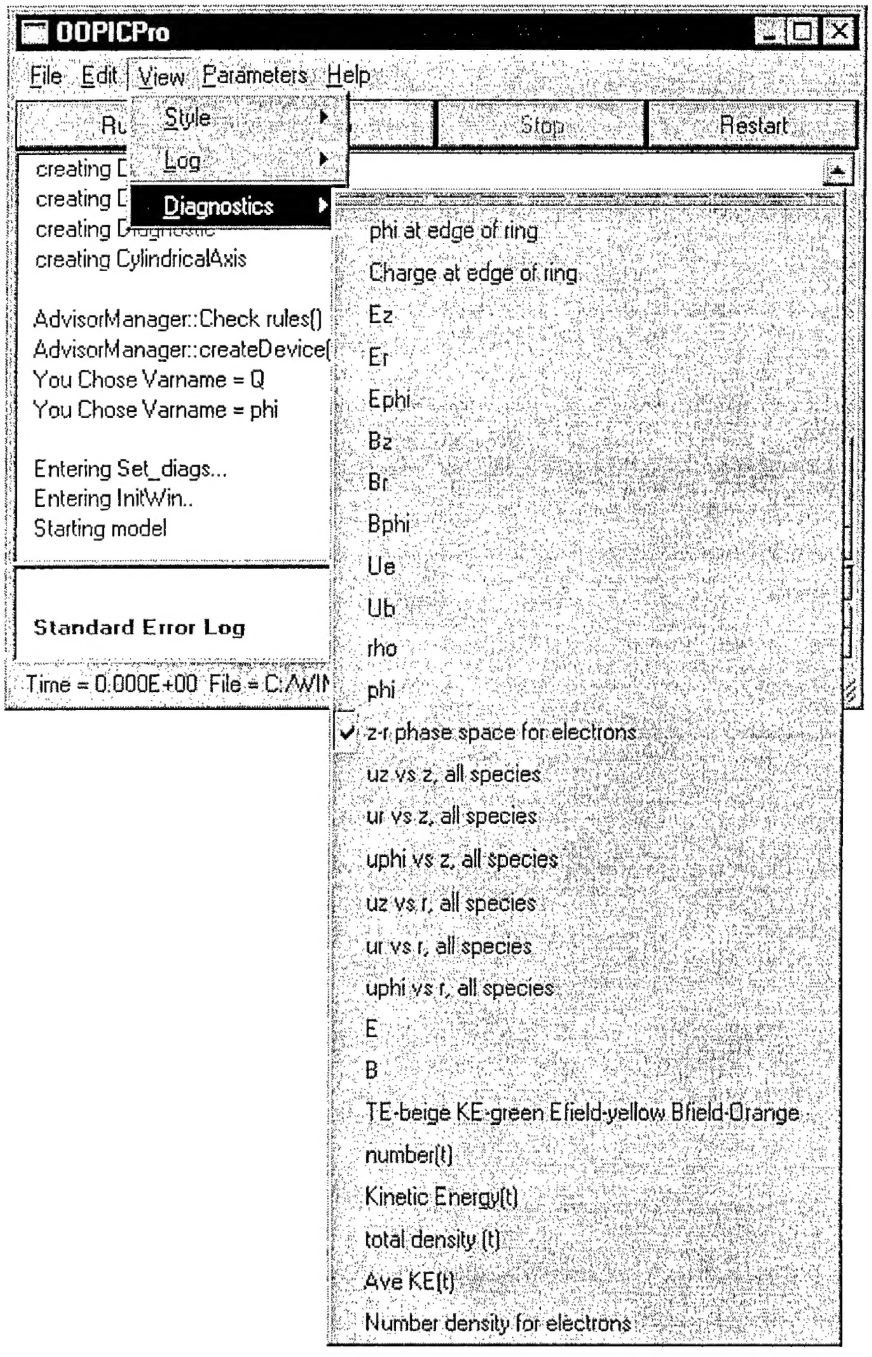
The next and last menu, the Help menu, allows the user to view various documentation for the program. One can access this manual by choosing Manual. Clicking on Software Documentation brings up code-specific documentation produced by Doxygen. Clicking the Text Documentation header causes the doc directory window to appear. It contains text files which were the documentation for OOPIC and XOOPIC, which can be viewed using any text viewer. All currently applicable information from the text files have been incorporated into this document. Copyright information can be viewed by clicking on the About heading.

VCR Control Buttons

The four large VCR buttons underneath the menu bar control the execution of the simulation. The [Run] button causes the simulation to run. The [Step] button advances the simulation one frame (note that this may not be the same as advancing the simulation one time step. The [Stop] button stops the simulation. The [Restart] button resets the simulation back to the initial conditions.

The diagnostic plots:

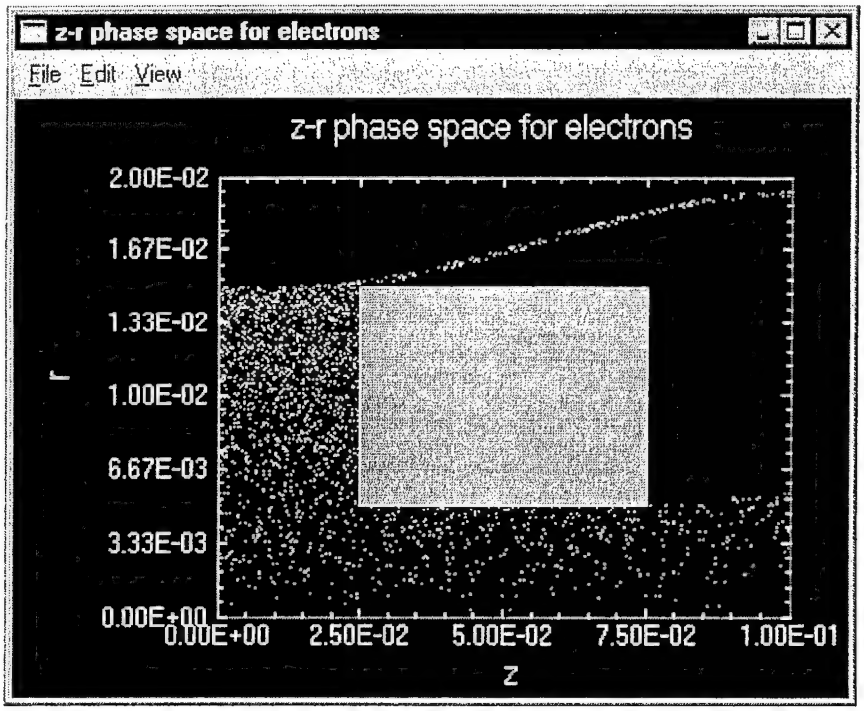
Upon opening OOPIC Pro, there will be position plots for each of the species present in the simulation. The user can bring up plots of other diagnostic parameters. First, open the View menu (click on View on the menu bar), hold the mouse arrow briefly over the word Diagnostic which appears with the rest of the menu (see the screen shot below). Then point the mouse arrow to the desired diagnostic and click again. A new window containing the plot will appear. Note: this can be done while the simulation is running. If the same diagnostic is selected a second time, the plot is removed.



There are four types of dignostic plots – particle plots, 3-D surface plots, line plots and vector plots.

Particle plots.

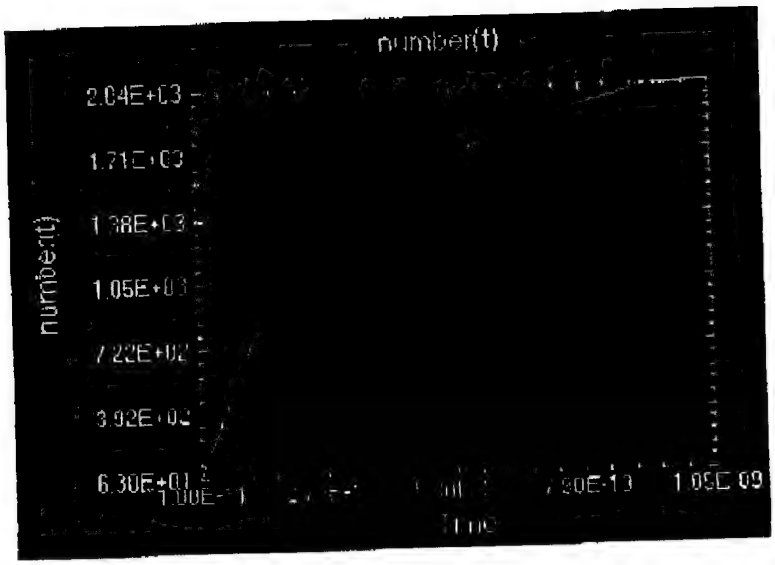
Particle plots include the configuration plots, which show the 2-D simulations region, with physical boundaries shown in various colors and the macroparticles of a given species shown as colored points. There is one configuration plot for each particle species – an example is provided by the screen shot below. There are also phase space particle plots, which show the macroparticle locations with a spatial coordinate along the horizontal axis and a momentum coordinate along the vertical axis. One plot for each combination of position and momentum is provided. In the phase space plots, the macroparticles for all species are shown simultaneously, differentiated by color. These colors are used consistently in all of the particle plots.



Like all of the 2-D plots in the OOPIC Pro GUI, the user can interactive change the axes limits, axes labels, plot title, etc. One can also zoom in to a smaller area of the plot, by choosing `Zoom in` from the `View` menu and then selecting the desired region with the mouse. After multiple zoom operations, the user can scroll back through previous views by clicking the left mouse button. At any point, the user can choose to apply the current settings and continue with the simulation.

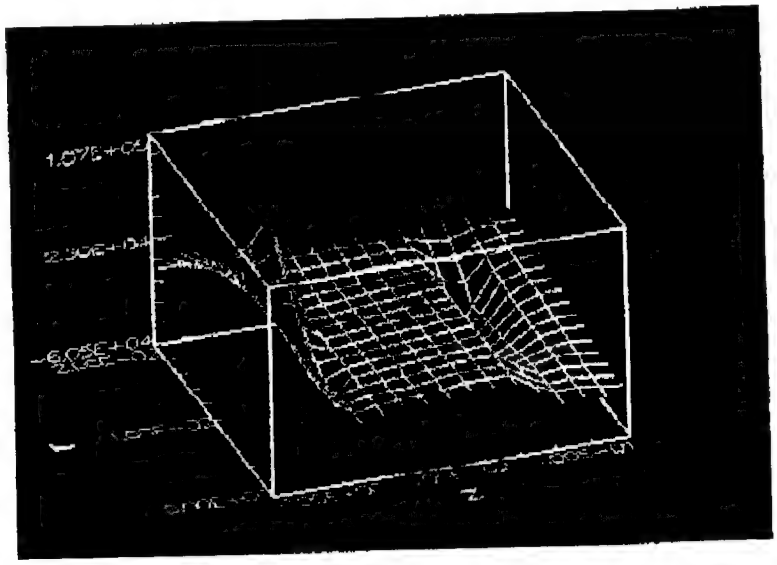
Line plots.

OOPIC Pro also provides a number of line plots, which show the time history of a specified quantity – an example is provided by the screen shot below, which shows the number of macroparticles in the simulation.



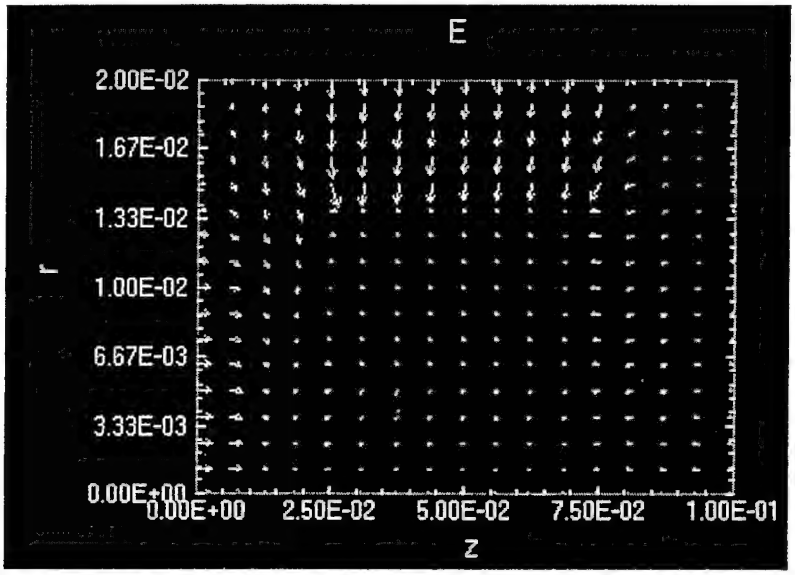
3-D surface plots.

OOPIC Pro provides many 3-D surface plots of quantities defined on the 2-D grid – an example is provided by the screen shot below, which shows the radial component of the electric field. In addition to all of the field components, one can plot particle densities, the density of any neutral gasses in the simulation, or FFT power spectral densities. These 3-D plots use the powerful and cross-platform OpenGL standard for fast 3-D rendering. The user can interactively rotate the data around any axis, and also zoom in or out



Vector plots.

OOPIC Pro also provides 2-D vector plots of quantities defined on the 2-D grid – an example is provided by the screen shot below, which shows the radial component of the electric field. In addition to all of the field components, one can plot particle densities, the density of any neutral gasses in the simulation, or FFT power spectral densities.



Saving and printing plots

Saving and printing are the same for all types of plots. To save a plot, select `Save as` from the File menu. In the window which appears, type a file name and choose a format in which to save the image. To print a plot, choose `Print` from the File menu.

Log windows

Both the Standard output log and the Standard error log are present at startup by default. The standard output log is in the upper part of the interior of the main control window. It displays comments on the status of the program, and gives messages as each component of an input file is interpreted and added to the simulation. The standard error file is in the lower part of the interior of the main control window.

Restart files

One can save the current state of a simulation in a cross-platform binary restart file. A restart file is in a block format where the specifications of the simulation and the current values of particle positions, fields, boundaries, and diagnostic values such as time histories, are written. To save a restart file, click on the File menu, choose the Save header, then click on Dump file. The default name for the dump file is the input file name with a .dmp extension.

One can also set OOPIC Pro to periodically save a restart file during a simulation. Under the Parameter menu, choose Iterations and click on Periodic Dump. Then change the number to be how many time steps are desired between file saves. The user can specify the name of the file.

To restore a simulation to the state at which it was when the restart file was saved, choose Open under the File menu and click on Dump file. Then open the directory containing the desired dump file and double click on the file. Note that although the simulation will be at the same state as before, GUI will not return to the previous state, since no information about the GUI is recorded in the dump file. Note: Currently, after loading a dump file, if one wants to completely restart with the initial conditions of the simulation, or to load another input file, OOPIC Pro must be exited and re-run.

The movie feature:

OOPIC Pro has the capability of saving multiple frames of one or more diagnostic plots during a simulation for later viewing as a movie. The following is a description of the process of making a movie.

First, in the Parameters menu, choose Movies, and then click on "Save Images as". By default, OOPIC Pro will place the image files in a directory named the same as the input file. If one is making a movie of a simulation for the first time, a box will pop up, informing the user that a new directory called <input file>_movie is being created, where<input file> is the name of the input file without the .inp extension. Hit OK, and a dialog window will appear. The default image file stub appears in the File Name input line. It is the input file name followed by an underscore. Change the file stub if desired. Then choose the format of the movie image files. The actual image file names will consist of the image file stub, followed by the name of the plot being saved, followed by the frame number, with an image file extension. For example, a typical file name would be "cavity_z-r_phase_space_for_electrons.00001.jpeg".

If the image file stub is not set by the user, the default is the input file name. The default directory is the oopicpro/input directory (not oopicpro/input/<input file name>_movie, and the default image format is '.ppm'.

After choosing an image file stub, the user must choose the frame rate of the movie. Again under the Parameters menu choose the Movies sub-menu, and click on `Iterations per Output.` The number in the entry window which appears specifies the number of simulation plot frames per movie frame. In a similar fashion, one can also specify the maximum number of frames in the movie.

If the simulation has multiple graphic plots open, a set of image files will be created from each plot window. Rather than load all of these files into the movie displaying program sequentially, the user can specify a Config file, which keeps track of the various plot movies.

Personnel Supported:

This project supported personnel from three institutions, including Tech-X Corporation, the Plasma Theory and Simulation Group (PTSG) at the University of California Berkeley, and a group at the University of Michigan.

Supported personnel at Tech-X included the principal investigator Dr. David Bruhwiler, as well as Kelly Luetkemeyer and Dr. John Cary.

Supported personnel at UC Berkeley included the principal investigator Prof. C. K. Birdsall, as well as Dr. John Verboncoeur and students Xingbo Yu and Peter Mardahl. Dr. Keith Cartwright completed his PhD thesis work with partial support from this project, and is now at the Air Force Research Laboratory (AFRL) at Kirtland AFB in New Mexico.

Supported personnel at the University of Michigan included the principal investigator Prof. Iain Boyd, as well as Dr. Jing Fan.

Publications:

See the "attachments" subsection below. Also, much of the work funded under this project has appeared in the recent text "High-Power Microwave Sources and Technologies" [1].

Attachments:

X. Yu, J.P. Verboncoeur and D.L. Bruhwiler, "A Radiation Damping Algorithm for Particle Simulation," unpublished report.

J.P. Verboncoeur and K.L. Cartwright, "Accuracy Analysis of the PIC Method", *Bull. Am. Phy. Soc.* **45** (2000).

D.L. Bruhwiler, "New Particle-in-Cell Algorithms for Improving Accuracy and Reducing Noise," presented at the AFOSR Electromagnetics Workshop (San Antonio, January, 2001).

D.L. Bruhwiler, K. Luetkemeyer, J.R. Cary, D.A. Dimitrov and P. Stoltz, "Error Analysis, Self-force Effects & Interactive Visualization for Particle-in-Cell Codes," presented at the Particle-in-Cell & Finite Difference Time Domain Workshop (Albuquerque, August, 2001).

J.P. Verboncoeur, "Symmetric spline weighting for charge and current density in particle simulation", accepted for publication in *J. Comp. Phys.* (2001).

J. P. Verboncoeur, "Digital filtering in radial coordinates for particle simulation", to be submitted to *J. Comp. Phys.* (2001).

I.D. Boyd and J. Fang, "Final Technical Report – Fluctuation Reduction Algorithms for Particle Simulations of Plasmas."

I.D. Boyd and J. Fang, "Fluctuation Reduction Algorithms for Particle Simulations of Plasmas" presented at the Particle-in-Cell & Finite Difference Time Domain Workshop (Albuquerque, August, 2001).

Interactions / Transitions:

The principal investigator Dr. David Bruhwiler and Prof. John Cary of the University of Colorado and Tech-X Corp. both attended the "AFOSR Electromagnetics Workshop," organized by Dr. Arje Nachman and held January 11-13 in San Antonio. Dr. Bruhwiler made a presentation [10], which is attached to this report.

The principal investigator Dr. David Bruhwiler and the PI's for the two subcontracts, Prof. Ned Birdsall and Prof. Iain Boyd, attended the "Particle-in-Cell / Finite Difference Time Domain" workshop, organized by Dr. Arje Nachman and Dr. Bob Peterkin and held August 21-22, 2001 at Kirtland AFB. Dr. Bruhwiler made a presentation [11], which is attached to this report. Prof. Boyd made a presentation [14], which is also attached to this report.

The most recent version of the XOOPIC code has been made available to Dr. Keith Cartwright of Kirtland AFB.

References

1. *High-Power Microwave Sources and Technologies*, ed. R.J. Barker and E. Schamiloglu (IEEE Press, New York, 2001).
2. X. Yu, J.P. Verboncoeur and D.L. Bruhwiler, "A Radiation Damping Algorithm for Particle Simulation," in preparation.
3. A.D. Yaghjian, *Relativistic Dynamics of a Charged Sphere: Updating the Lorentz-Abraham Model* (Springer-Verlag, New York, 1992).
4. R.W. Ziolkowski and A.D. Yaghjian, "Upgrading the Field-Particle Physics and Numerics of PIC codes for HPM Tube Design," STTR Final Report to AFOSR (1999), unpublished.
5. J.D. Jackson, *Classical Electrodynamics*, John Wiley and Sons (1975).
6. S.T. Perkins, D.E. Cullen and S.M. Seltzer, "Tables and Graphs of Electron-Interaction Cross Sections from 10 eV to 100 GeV," LLNL report UCRL-50400, Vol. 31.
7. K.L. Cartwright, J.P. Verboncoeur, and C.K. Birdsall, "Loading and Injection of Maxwellian Distributions in Particle Simulations", *J. Comp. Phys.* **162**, 483-513 (2000).
8. C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulation*, (McGraw-Hill, New York, 1985).
9. J.P. Verboncoeur and K.L. Cartwright, "Accuracy Analysis of the PIC Method", *Bull. Am. Phy. Soc.* **45** (2000).
10. D.L. Bruhwiler, "New Particle-in-Cell Algorithms for Improving Accuracy and Reducing Noise," presented at the AFOSR Electromagnetics Workshop (San Antonio, January, 2001).
11. D.L. Bruhwiler, K. Luetkemeyer, J.R. Cary, D.A. Dimitrov and P. Stoltz, "Error Analysis, Self-force Effects & Interactive Visualization for Particle-in-Cell Codes," presented at the Particle-in-Cell & Finite Difference Time Domain Workshop (Albuquerque, August, 2001).
12. J.P. Verboncoeur, "Symmetric spline weighting for charge and current density in particle simulation", accepted for publication in *J. Comp. Phys.* (2001).
13. J.P. Verboncoeur, "Digital filtering in radial coordinates for particle simulation", to be submitted to *J. Comp. Phys.* (2001).
14. I.D. Boyd and J. Fang, "Final Technical Report – Fluctuation Reduction Algorithms for Particle Simulations of Plasmas."
15. I.D. Boyd and J. Fang, "Fluctuation Reduction Algorithms for Particle Simulations of Plasmas" presented at the Particle-in-Cell & Finite Difference Time Domain Workshop (Albuquerque, August, 2001).
16. G.A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, (Oxford Univ. Press, Oxford, 1994).
17. C.-P. Cai, J. Fan, I.D. Boyd and G.V. Candler, "Direct Simulation Methods for Micro-Channel Flows," AIAA Paper 99-3801 (June, 1999).
18. J. Fan, I.D. Boyd, C.-P. Cai, K. Hennighausen and G.V. Candler, "Computation of Rarefied Gas Flows Around a NACA 0012 Airfoil," AIAA Paper 99-3804 (June, 1999).

Error Analysis, Self-force Effects & Interactive Visualization for Particle-in-Cell Codes

David Bruhwiler
with
Kelly Luetkemeyer, John Cary, Dimitre Dimitrov, Peter Stoltz

Presented at:
Particle-in-Cell & Finite Difference Time Domain Workshop
Kirtland AFB -- Albuquerque NM -- August 21 & 22, 2001

Tech-X Corporation
5541 Central Ave., Suite 135
Boulder, Colorado 80301
http://www.techxhome.com

Acknowledgements

- STTR from the Air Force Office of Scientific Research
  - Dr. Arje Nachman (contract monitor)
  - contract # F49620-99-C-0028
- Tech-X Corporation
  - David Bruhwiler (principal investigator)
  - Kelly Luetkemeyer, John Cary, Dimitre Dimitrov & Peter Stoltz
- University of California Berkeley (STTR partner)
  - Ned Birdsall (principal investigator)
  - John Verboncoeur, Keith Cartwright, Peter Mardahl, Xingbo Yu
- University of Michigan (subcontract)
  - Iain Boyd (principal investigator)
  - Jing Fan
- Air Force Research Laboratory (Kirtland AFB)
  - Keith Cartwright

Overview

Three consecutive presentations to describe this work:

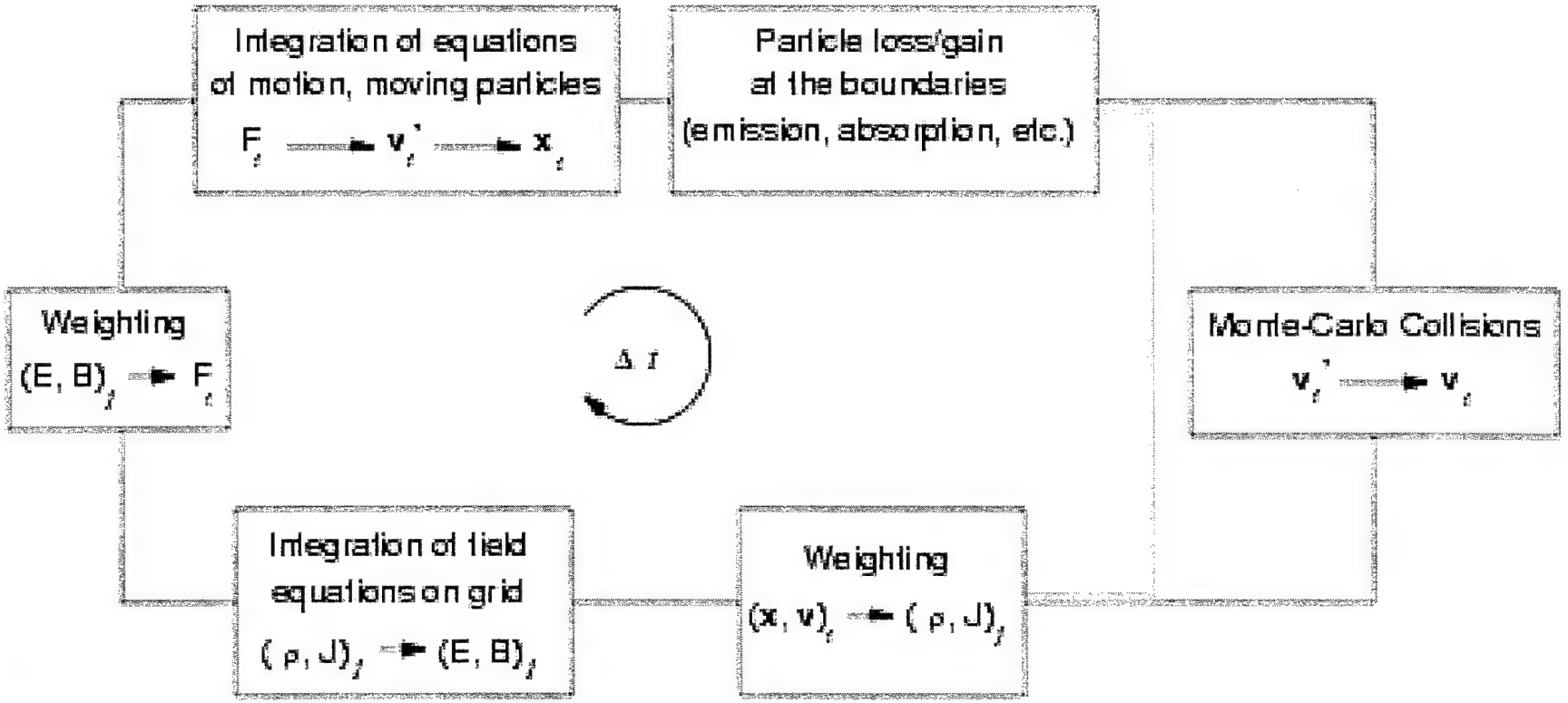
- David Bruhwiler / Tech-X Corporation
  - analysis of errors in the fundamental 2-D PIC algorithm
  - analysis of the self-force for relativistic electrons
  - cross-platform graphical user interface (GUI)
- Ned Birdsall / University of California Berkeley
  - improved charge and current weighting
  - noise reduction via digital filtering
  - treatment of secondary electrons emitted from metal surfaces
- Iain Boyd / University of Michigan
  - application of information preservation techniques to PIC

Outline

- Analysis of errors in the fundamental 2-D PIC algorithm
  - methodology
  - electrostatic case
  - electromagnetic case
- Analysis of self-force effects for PIC
  - consideration of a fully-consistent model
  - implementation and testing of a simplified model
  - consideration of self-force effects during collisions
- Interactive visualization of PIC simulations
  - cross-platform GUI (MS Windows, Unix/Linux *& Mac OS X*)
  - uses OpenGL for fast 3-D rendering
- Commercialization is a central goal of the STTR program
  - the GUI is central to our plans for commercializing OOPIC
  - free noncommercial use on Unix/Linux benefits PIC community

Error Analysis of the Generic PIC Algorithm

- Work done primarily by K. Cartwright & J. Verboncoeur
- Motivation
  - wanted a coherent treatment of the full PIC loop
  - attempted to capture interaction of point particles with gridded fields

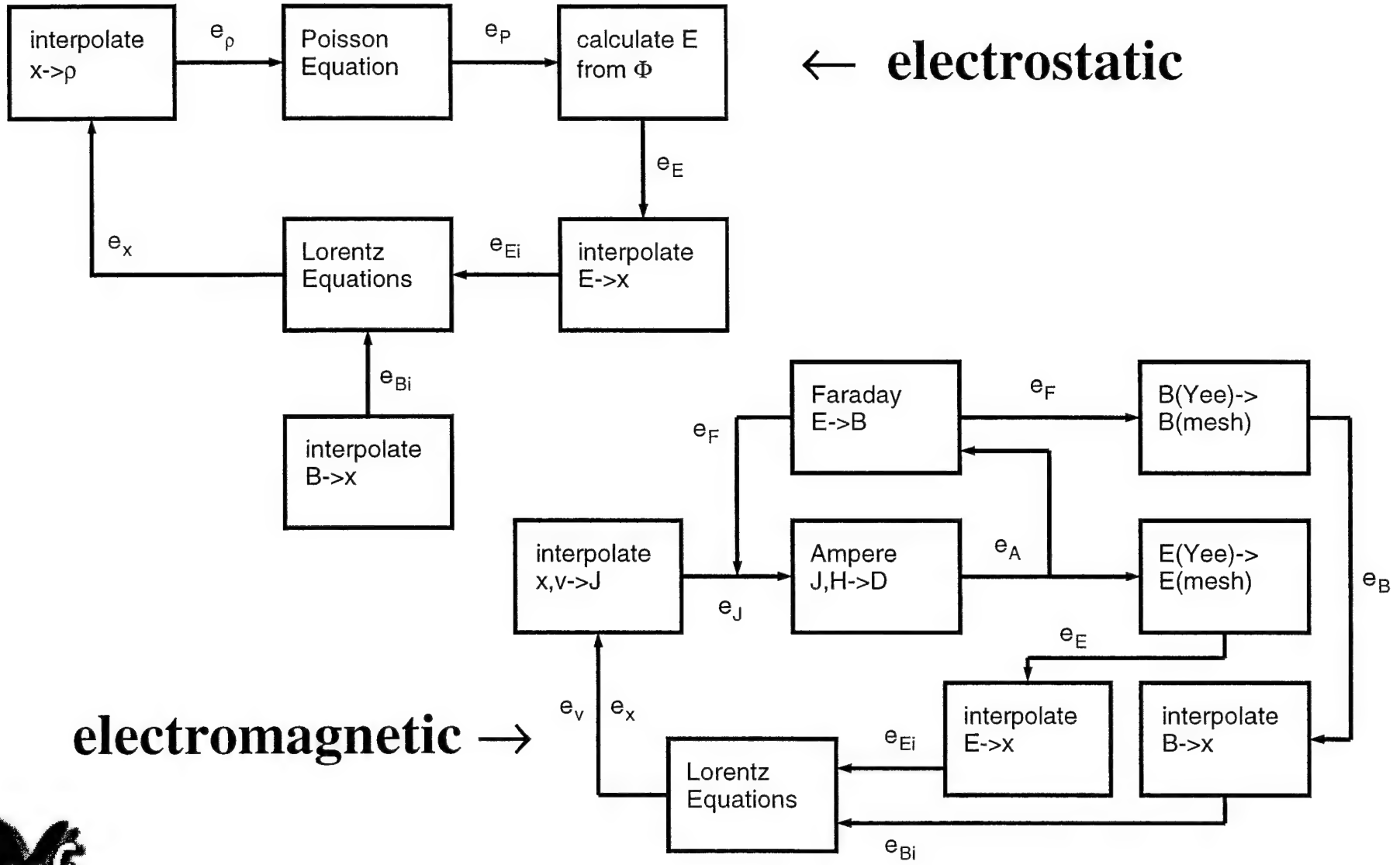


Previous Work & Chosen Methodology

- Separate components of the electrostatic PIC algorithm have been analyzed previously
  - by Birdsall & Langdon and Hockney & Eastwood
- Other aspects of the problem were not previously addressed
  - Integrated analysis was not done previously (to our knowledge)
  - Very little analysis of the electromagnetic case is available

- Methodology for extending previous work to full PIC loop:
  - must propagate errors from one algorithm to the next
  - must work in the limit of many macro-particles per cell
    - » thus, noise due to finite particle statistics is not captured
  - enables coherent, comprehensive error analysis

Error Flow Schematics



Mathematica is Used to Handle the Algebraic Complexity

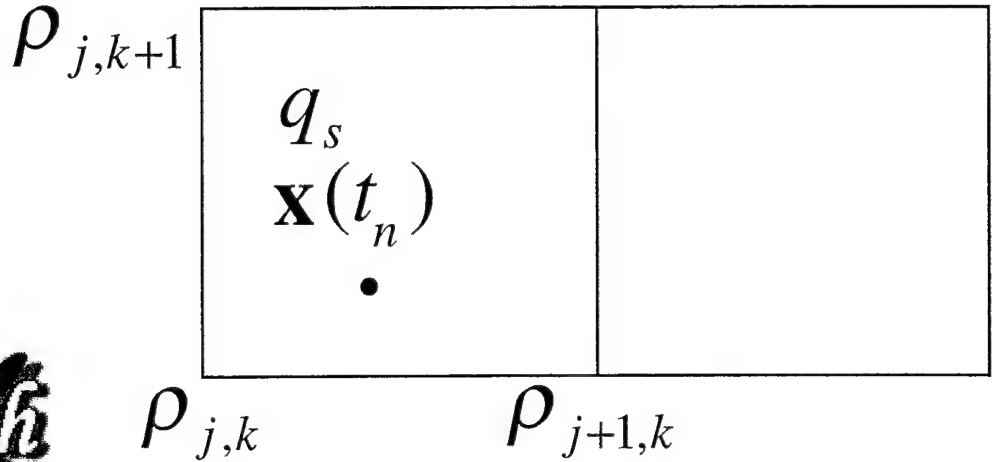
- Finite difference errors calculated via Taylor expansions
  - requires a tremendous amount of algebra in multiple dimensions
- Mathematica allows for treatment of general cases:
  - different grid shapes (square vs. rectangular)
  - different shape functions for weighting particles to grid
  - different interpolation schemes for weighting fields to particles
- Easy to collaborate via sharing of Mathematica files
  - Keith Cartwright began this work as a student in Berkeley
  - Peter Stoltz is using Keith's files to continue the work

Charge Deposition for Electrostatic PIC

- Cannot treat individual particles
  - must assume many macro-particles per grid cell
- The "correct" charge distribution is some ideal $\rho(x)$
  - obtained in principle by integrating over the particle distribution
- The deposited charge $\rho_{j,k}$ depends on the shape function
  - essentially, Taylor expand ρ and integrate over the shape function

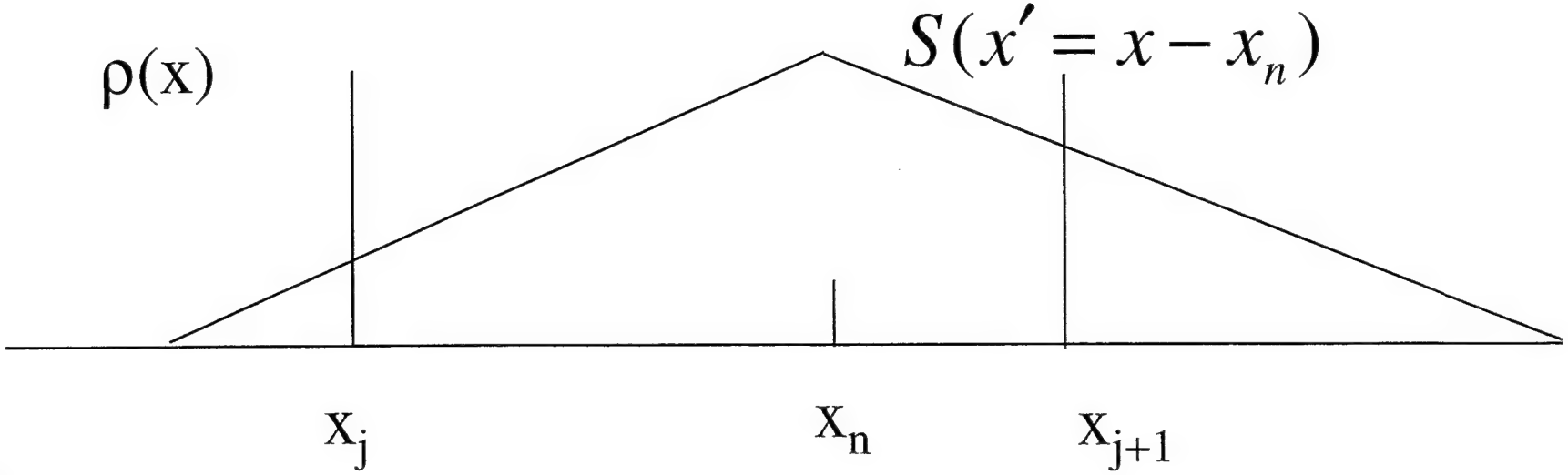
$$\rho(x,y) \approx \rho(x_0,y_0) + (x-x_0)\frac{\partial \rho}{\partial x}(x_0,y_0) + (y-y_0)\frac{\partial \rho}{\partial y}(x_0,y_0) + \ldots$$

- The error is the difference between the two: $e_\rho = \rho(x_{j,k}) - \rho_{j,k}$



Shape Function determined by Deposition Algorithm

- Shape function for linear charge deposition is shown below
- Odd terms in Taylor expansion killed by symmetry
- NGP, linear and 2nd-order shape fcn's yield 2nd-order errors
  - coefficient actually increases slowly with higher order
  - because higher-order algorithms smear the distribution more
  - this neglects fact that higher-order helps to quiet statistical noise



Cancellation of Errors can Occur for Linear Deposition

- Error due to finite difference of Poisson's equation cancels with error due to charge weighting with linear shape function.

$$e_{\rho,NGP} \approx \frac{\Delta x^2}{12}\rho^{(2)} + \ldots$$

$$e_{\nabla^2\Phi,two-point} \approx \frac{\Delta x^2}{12}\Phi^{(4)} + \ldots$$

  - Coefficients of error terms are the same to all orders, and so combined with $\rho \sim \Phi^{(2)}$, one gets error cancellation
  - Poisson's equation does not mix space and time, nor does it mix different spatial directions

Errors in the Full Electrostatic PIC Loop

- One major surprise:
  - truncation errors in the solution of Poisson's equation are exactly cancelled by errors for linear weighting of charge density
    - » TO ALL ORDERS
  - higher-order charge weightings do not have this nice property
  - of course, other 2^{nd}-order errors remain
- Errors in the full loop are 2^{nd}-order in both space and time
  - this is true for each step of the analysis
  - no mechanism for errors to "feed down" to lower order
  - our contribution was to combine analysis of particles and fields

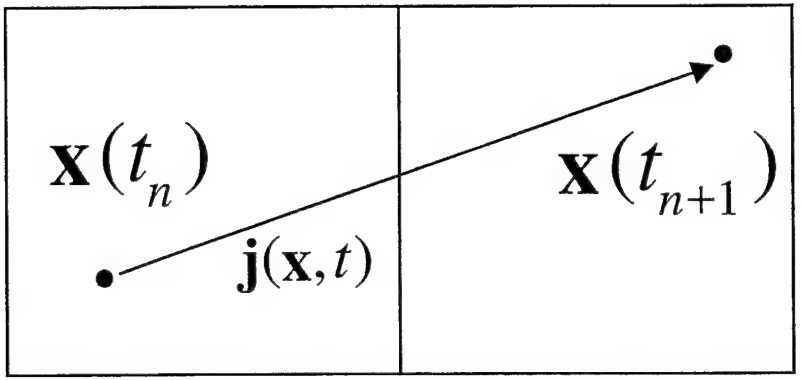
Error Analysis for the Electromagnetic PIC Loop

- Analogous to the electrostatic case
  - no fundamental difference in the approach
  - but everything is more complicated and much messier
- Maxwell's equations couple x and y (2D PIC code)
  - whereas Poisson's equation is separable
- Relativistic γ factor couples all velocity components
- Staggered field positions on Yee mesh
  - leads to several interpolation functions
- Flow of the error analysis is more complicated
- Current deposition is more complicated than charge dep.
  - charge conservation requires asymmetry (e.g. NGP-linear)
  - difficult to incorporate particles that cross grid lines

Current Deposition for Electromagnetic PIC

- Handled in a manner directly analogous to charge deposition
  - but grid crossings are difficult to handle
- Charge conservation requires asymmetric weighting
  - NGP along direction of current, linear weighting transversely
  - bilinear current deposition requires correction schemes
  - we will only consider NGP-linear, which is very common

$$\mathbf{j}(\mathbf{x}) \approx \mathbf{j}(\mathbf{x}') + \mathbf{j}^{(1)}(\mathbf{x}')(\mathbf{x}-\mathbf{x}') + \frac{1}{2}\mathbf{j}^{(2)}(\mathbf{x}')(\mathbf{x}-\mathbf{x}')^2$$



Some fraction of current goes to each cell. Any error due to this must be handled separately.

Error Cancellation not Possible for Electromagnetic Case

- Error in finite difference solution to Ampere's Law involve time and space and mixes different spatial directions:

$$e_{j_x,\text{NGP-linear}} \approx \frac{\Delta y^2}{12} j_x^{(0,2)} + \frac{\Delta x^2}{24} j_x^{(2,0)} + \ldots$$

$$e_{\text{Ampere}_x,\text{two-point}} \approx \frac{\Delta t^2}{24} D_x^{(3)} + \frac{\Delta y^2}{24} H_z^{(0,3)} + \ldots$$

  - Ampere's Law mixes spatial and temporal derivatives, making total error cancellation impossible

Error in Newton-Lorentz equation goes like γ^6

- Newton-Lorentz equation:

$$m\frac{d}{dt}(\gamma\mathbf{v}) = q(\mathbf{E} + \mathbf{v}\times\mathbf{B})$$

- Solving this on a grid with a Boris integration scheme*:

$$e_{Lorentz} \sim \gamma^6 \Delta t^2$$

  - For ultra-relativistic simulations, this could become a problem

**result provided by Keith Cartwright*

Analysis of the Self-Force Effect for Electrons

- Self-force effects cannot be directly modeled via PIC
  - effects are included (in principle) in the limit $\Delta t, \Delta x \rightarrow 0$
  - practical values for $\Delta t, \Delta x$ are many orders of magnitude too large
- Complete analytical treatment of self-force is available
  - Yaghjian, *Relativistic Dynamics of a Charged Sphere* (S-V, 1992)
- Self-force shown to be negligible for microwave devices
  - Ziolkowski & Yaghjian (1999), unpublished report.

- We developed & implemented a simplified treatment
  - relevant to ultra-relativistic electrons in a strong magnetic field
  - back reaction on electron due to classical synchrotron radiation
- We explored self-force in the context of particle collisions
  - also shown to be negligible for evacuated microwave devices

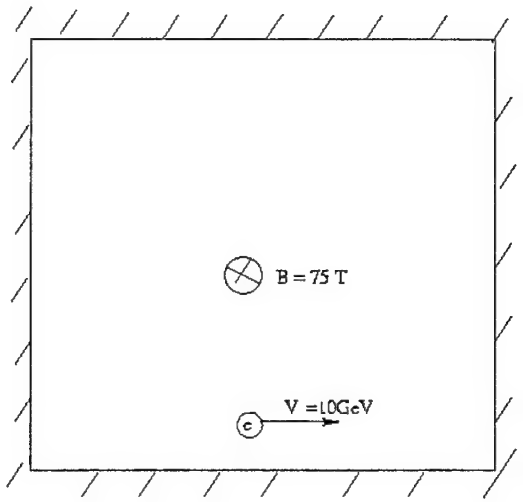
Simplified Treatment of the Self-Force

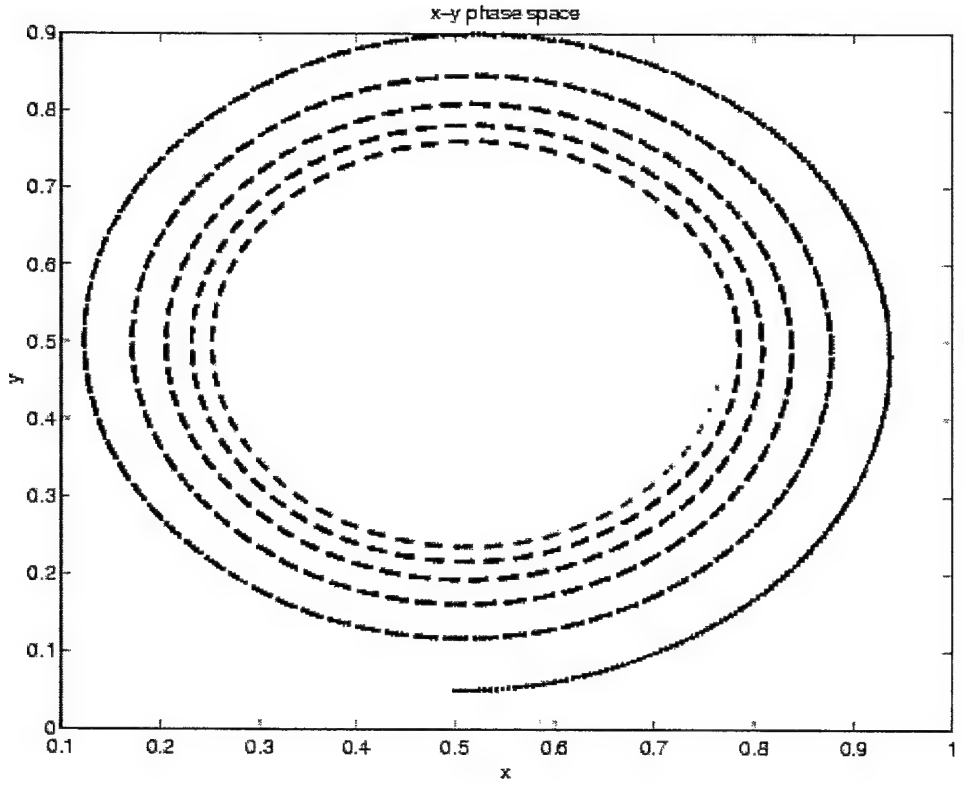
- Complete equations are not practical for implementation
  - very complicated and so would be computationally expensive
  - requires 2nd time-derivative of momentum, which would be noisy
- In physical limit where self-force effects are important
  - classical recoil due to synch. radiation is reasonable approximation
  - relevant equations are simple & can be found in Jackson
  - requires only the 1st time-derivative of the momentum
- Change in normalized momentum $u=p/m=\gamma v$:

$$\left(\frac{du}{dt}\right)_{rad} = -\frac{1}{6\pi\varepsilon_0}\frac{e^2}{mc^5}\gamma^2\left[\left(\frac{d\mathbf{u}}{dt}\right)^2 - \beta^2\left(\frac{du}{dt}\right)^2\right]$$

Testing Self-Force Implementation in XOOPIC

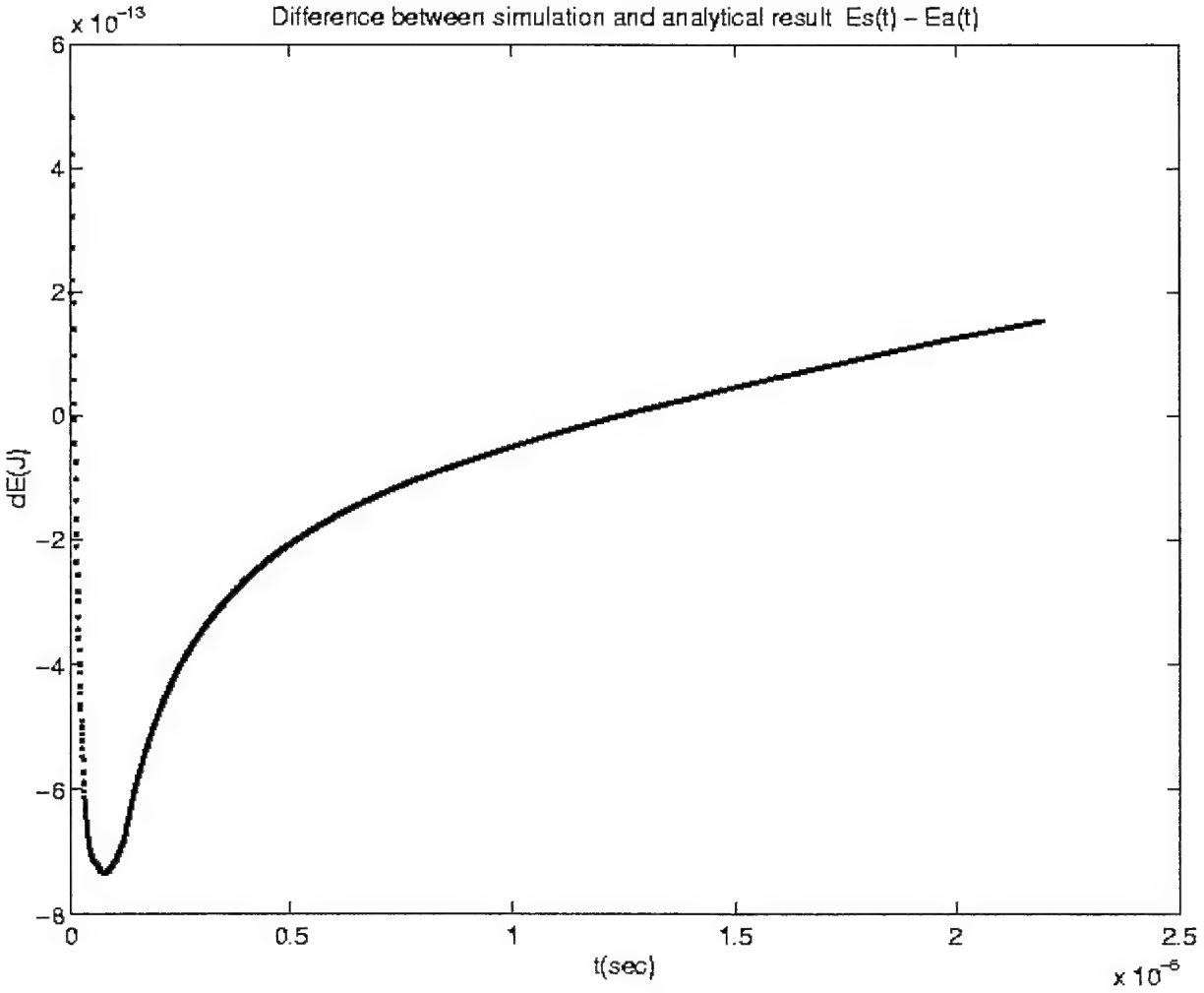
- 10 GeV electron in a 75 Tesla magnetic field
  - radius of cyclotron orbit decreases due to synchrotron energy loss





Testing Self-Force Implementation in XOOPIC

- 10 GeV electron in a 75 Tesla magnetic field
  - energy loss of single particles agrees well with theory



Self-Force Effects Due to Collisions

- Collisions between electrons and background neutrals
  - generates bremsstrahlung, which reduces electron energy
  - best handled as a collisional process
  - XOOPIC includes code for handling MCC algorithms
  - 1991 LLNL report presents relevant energy loss data
    - » effect is negligible for evacuated microwave devices

Interactive GUI for Particle-in-Cell Codes

- A major component of the work at Tech-X
  - original idea was a new GUI for XOOPIC
  - developed a scientific graphics package
    - » well suited for particle-in-cell code
    - » also being used for high-energy particle accelerator design codes
    - » suitable for any application that needs interactive visualization
- Chose to use C++
  - XOOPIC is written in C++
  - Tech-X is a big believer in object-oriented programming
  - we found an emerging C++ windowing toolkit, Qt
- Cross-platform was considered essential
  - high-performance UNIX workstations for state-of-the-art
  - MS Windows for commercial viability

Qt -- a cross-platform C++ windowing toolkit

- Developed by Troll Tech Inc. of Norway
  - http://www.trolltech.com
- Open source software on Unix with large user base
  - open source KDE project (desktop for linux) is based on Qt
  - this will allow OOPIC to remain free on all Unix platforms
- Commercial license available for Win32 & Unix
  - enables commercial version of OOPIC for PC's
  - Troll Tech is now porting Qt to Mac OS X
- Pure C++ toolkit with a good OO design
  - allows for seamless integration with OOPIC
  - enables design/implementation of good scientific graphics
- Cross-platform (single interface; two implementations)
  - same source code for both PC and Unix versions of XOOPIC
  - develop on PC, then build on Unix (or vice-versa)

Use of OpenGL for 3-D Rendering

- Qt provides good support for 2-D scientific graphics
- For 3-D graphics, Qt works seamlessly with OpenGL
  - OpenGL calls go directly to the native API with no overhead
  - Qt provides a nice window around the OpenGL rendering
- OpenGL is a low-level API (interface) for 3-D graphics
  - originally developed by SGI; their hardware is optimized for it
  - implementations exist for all platforms (Win32, Unix, Mac)
  - many accelerator graphics cards are available (games)
  - now the standard for high-performance 3-D visualization
- Linux is a special case, but not for long:
  - commercial drivers for accelerator cards are now available
  - SGI made GLX (OpenGL/X11 interface) open source for Xfree86
    - » so, free implementations and drivers will become available...
  - Mesa is a free implementation for linux (not officially OpenGL)

Qt-based GUI Prototype is Complete

- Multiple 2-D plots of real space and velocity space can be opened and closed dynamically during a simulation
- Multiple 3-D plots of density, field components, etc. have been implemented via OpenGL
- Prototype works on many platforms:
  - Win 98/NT, Solaris, PC/linux, Dec Alpha/linux, SGI/Irix
  - identical look and feel on all platforms, with the same source code

- OOPIC Pro is ready for beta testing
- Graphics package will also be released independently
  - free-for-noncommercial-use on Unix
  - commercial only for MS Windows and (eventually) Mac OS X

New Particle-in-Cell Algorithms for Improving Accuracy and Reducing Noise

David Bruhwiler

Presented at the:

AFOSR Electromagnetics Workshop
San Antonio, Texas -- January 11-13, 2001

Tech-X Corporation
5541 Central Ave., Suite 135
Boulder, Colorado 80301
http://www.techxhome.com

Acknowledgements

- STTR from the Air Force Office of Scientific Research
  - Dr. Arje Nachman
  - contract # F49620-99-C-0028
- Tech-X Corporation
  - David Bruhwiler
  - Kelly Luetkemeyer
  - John Cary
- University of California Berkeley
  - Ned Birdsall
  - John Verboncoeur
  - Peter Mardahl
  - Xingbo Yu
- Air Force Research Laboratory (Kirtland AFB)
  - Keith Cartwright

Overview

- Developing improved algorithms for PIC codes
  - analysis of the self-force for relativistic electrons
  - treatment of secondary electrons emitted from metal surfaces
  - improved models for particle loading and injection
  - analysis of errors in the fundamental 2-D PIC algorithm
  - methods for reducing noise
  - electromagnetic boundary conditions
- The XOOPIC particle-in-cell code
  - used in this project for implementation & testing of algorithms
  - used at AFRL/Kirtland by K. Cartwright
    - » useful for 2-D simulations of HPM pulse shortening in a MILO
- Tech-X developing GUI for a commercial OOPIC version
  - cross-platform (MS Windows, Unix, Linux)
  - uses OpenGL for fast 3-D rendering

Analysis of the Self-Force Effect for Electrons

- Self-force effects cannot be directly modeled via PIC
  - effects are included (in principle) in the limit $\Delta t, \Delta x \rightarrow 0$
  - practical values for $\Delta t, \Delta x$ are many orders of magnitude too large
- Complete analytical treatment of self-force is available
  - Yaghjian, *Relativistic Dynamics of a Charged Sphere* (S-V, 1992)
- Self-force shown to be negligible for microwave devices
  - Ziolkowski & Yaghjian (1999), unpublished report.

- We developed & implemented a simplified treatment
  - relevant to ultra-relativistic electrons in a strong magnetic field
  - back reaction on electron due to classical synchrotron radiation
- We explored self-force in the context of particle collisions
  - also shown to be negligible for evacuated microwave devices

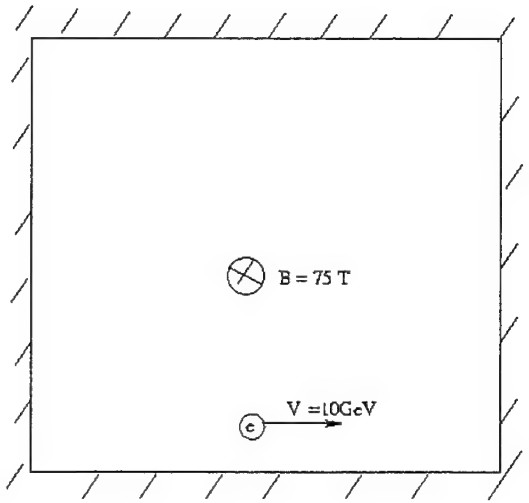
Simplified Treatment of the Self-Force

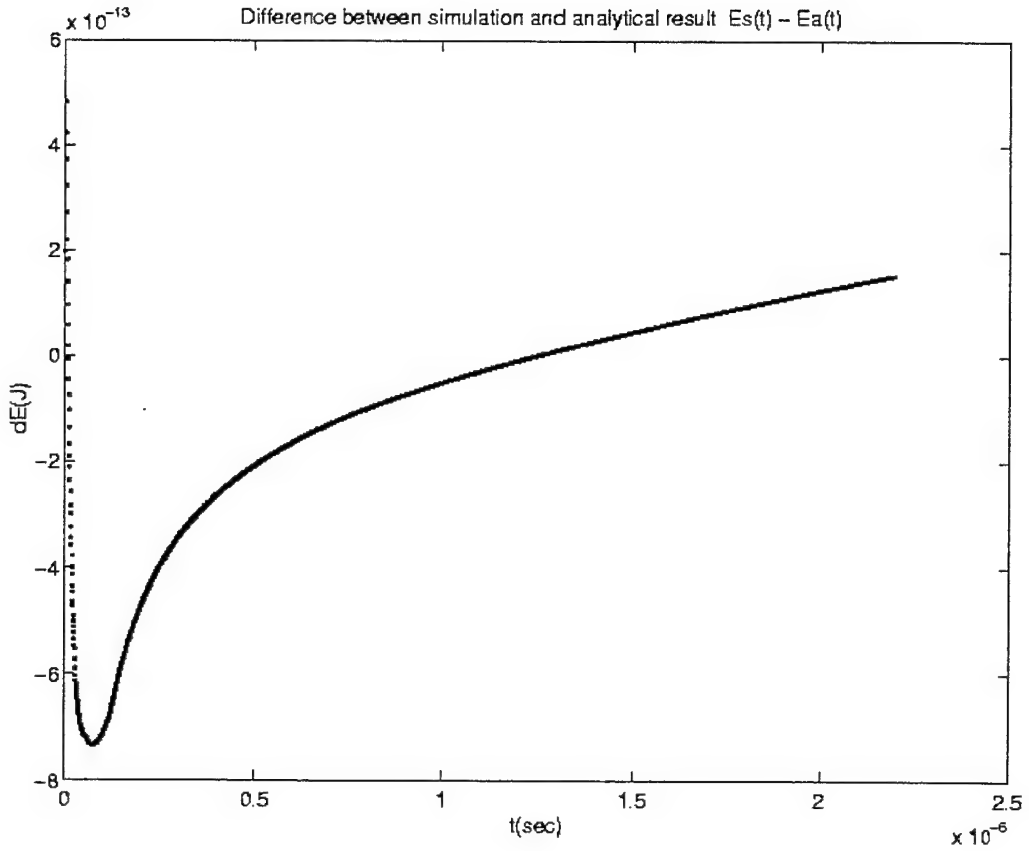
- Complete equations are not practical for implementation
  - very complicated and so would be computationally expensive
  - requires 2nd time-derivative of momentum, which would be noisy
- In physical limit where self-force effects are important
  - classical recoil due to synch. radiation is reasonable approximation
  - relevant equations are simple & can be found in Jackson
  - requires only the 1st time-derivative of the momentum
- Change in normalized momentum u=p/m=γv:

$$\left(\frac{du}{dt}\right)_{rad} = -\frac{1}{6\pi\varepsilon_0}\frac{e^2}{mc^5}\gamma^2\left[\left(\frac{d\mathbf{u}}{dt}\right)^2 - \beta^2\left(\frac{du}{dt}\right)^2\right]$$

Testing Self-Force Implementation in XOOPIC

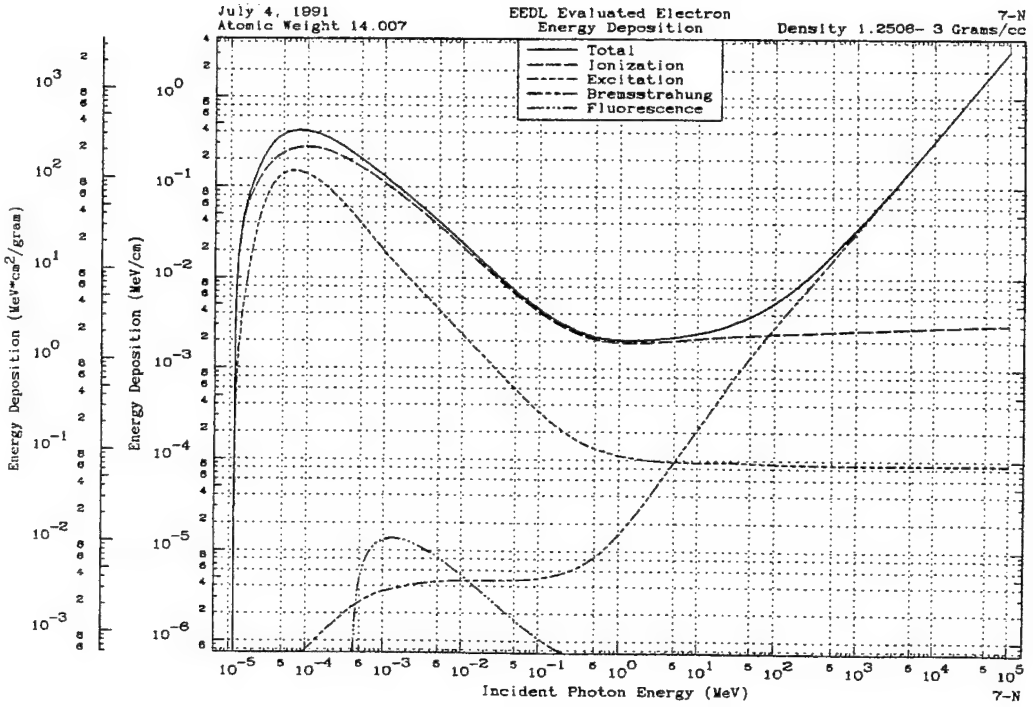
- 10 GeV electron in a 75 Tesla magnetic field
  - energy loss of single particles agrees well with theory





Self-Force Effects Due to Collisions

- Collisions between electrons and background neutrals
  - generates bremsstrahlung, which reduces electron energy
  - best handled as a collisional process
  - XOOPIC includes code for handling MCC algorithms
  - 1991 LLNL report presents relevant energy loss data
    - » effect is negligible for evacuated microwave devices



Emission of Secondary Electrons

- Accurate simulation of secondary emission aids in:
  - optimizing depressed collectors to maximize tube efficiency
  - understanding multipactor discharge on vacuum windows
  - beam interception, particle loss to walls, etc.
- Model for secondary coefficient:
  - based on previous work by Vaughan, Shih, Gopinath
  - includes energy and angle of primary electron:

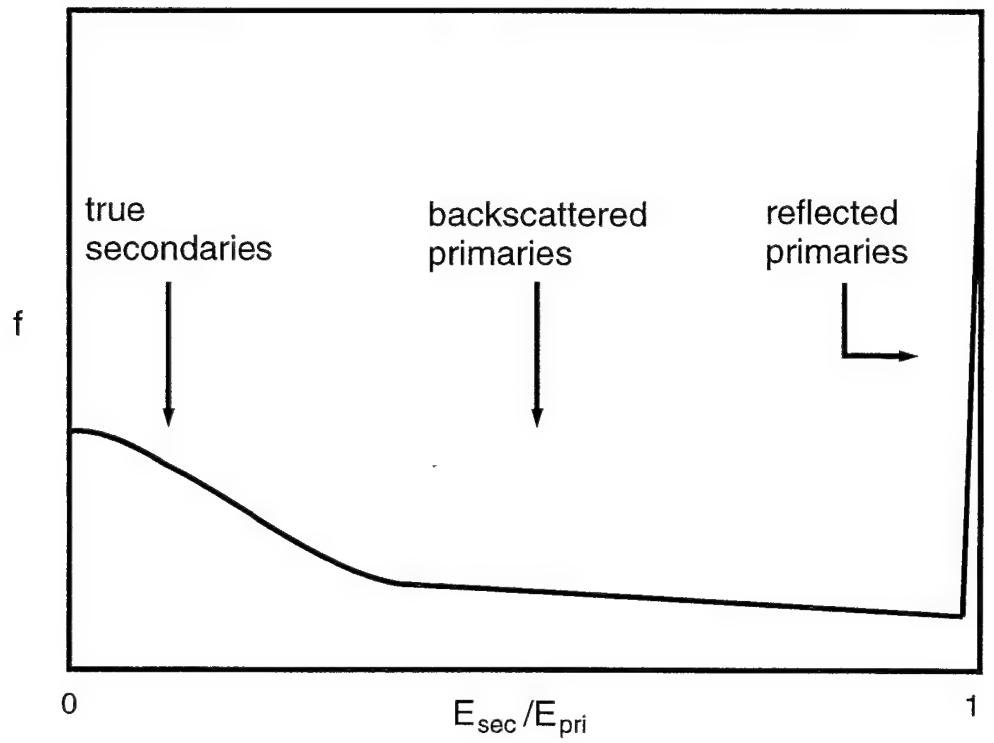
$$\sigma(V,\theta) = \delta_{\max 0}\left(1 + k_s \frac{\theta^2}{2\pi}\right)\left(w \exp(1-w)\right)^k,$$

  - energy dependence appears implicitly via

$$w = \frac{V - V_0}{V_{\max 0}\left(1 + k_s \theta^2 / 2\pi\right) - V_0} \qquad k = \begin{cases} 0.62 & w \le 1 \\ 0.25 & w > 1 \end{cases}$$

Secondary Emission Spectrum

- Model includes angle/energy distribution of secondaries
  - based on previous work by Spangenberg and Vaughan
- There are three types of secondaries
  - true secondaries
  - backscattered primaries
  - reflected primaries



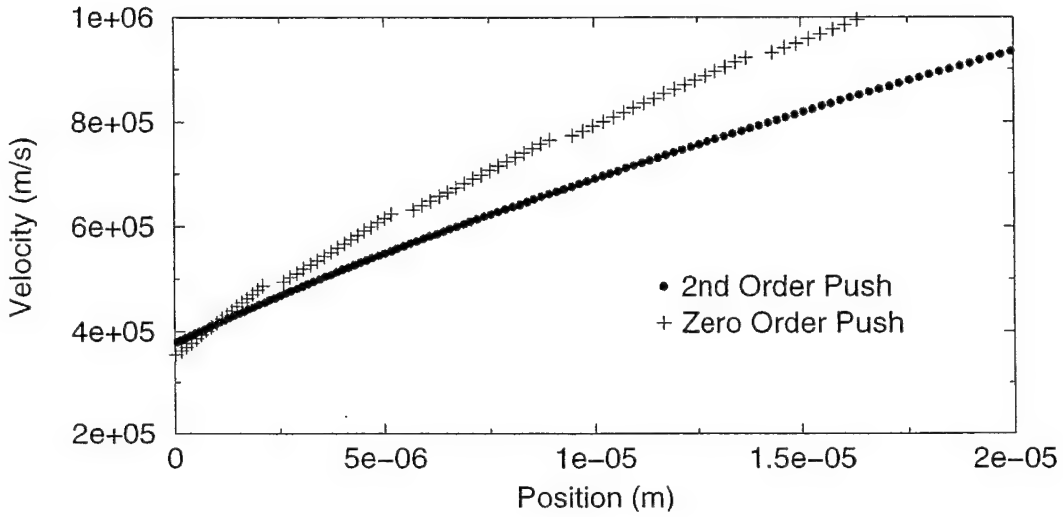
Implementation & Use of the Secondary Model

- The model has been implemented in two codes
  - XOOPIC
  - XPDP1
    - » a 1-D PIC code developed at UC Berkeley

- 1-D simulations of multipactor discharge on a window
  - Valfells, Verboncoeur & Lau, IEEE Trans. Plasma Sci., **28** (2000)

Particle Loading and Injection

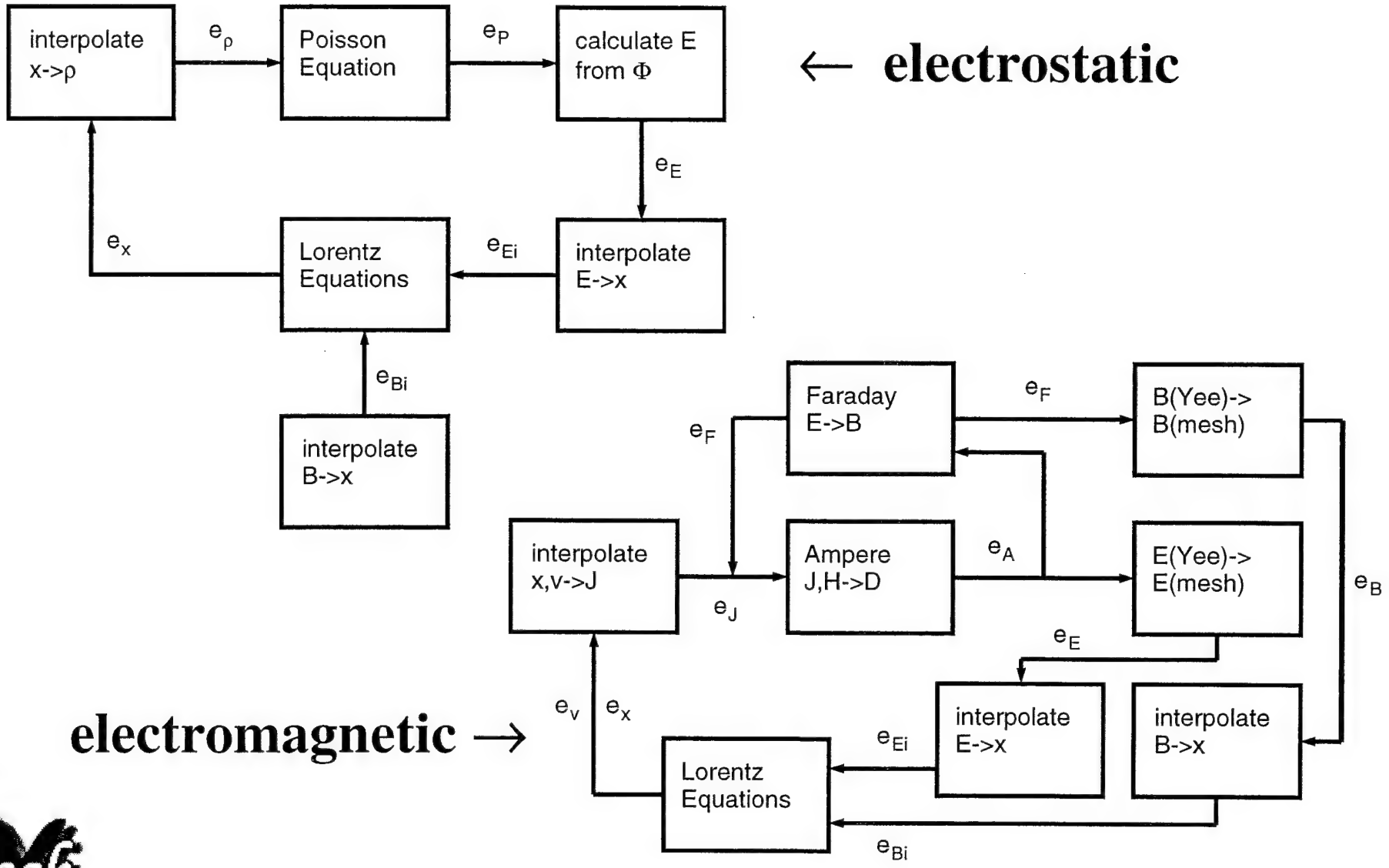
- Particle injection can significantly impact simulation results
  - time centering of position & velocity is necessary
  - else, can break 2nd-order accuracy of particle integrator
  - else, can modify E-field at emitting surface
  - Cartwright, Verboncoeur, Birdsall, J. Comp. Phys. (2000), accepted.



Error analysis of Generic PIC Algorithm

- Errors in separate components of the electrostatic PIC analysis have been analyzed previously
  - Birdsall & Langdon, Hockney & Eastwood
  - Integrated analysis was not done previously (to our knowledge)
  - Analysis has not yet been applied to electromagnetic case
- We have extended work of B&L, H&E
  - propagated errors from one algorithm to the next
    - » provides complete error analysis of generic PIC algorithm
    - » relevant as "initial condition" for electromagnetic PIC
- One major surprise:
  - truncation errors in the solution of Poisson's eqn are exactly cancelled by errors for linear weighting of charge density
    - » TO ALL ORDERS
  - higher-order charge weightings do not have this nice property

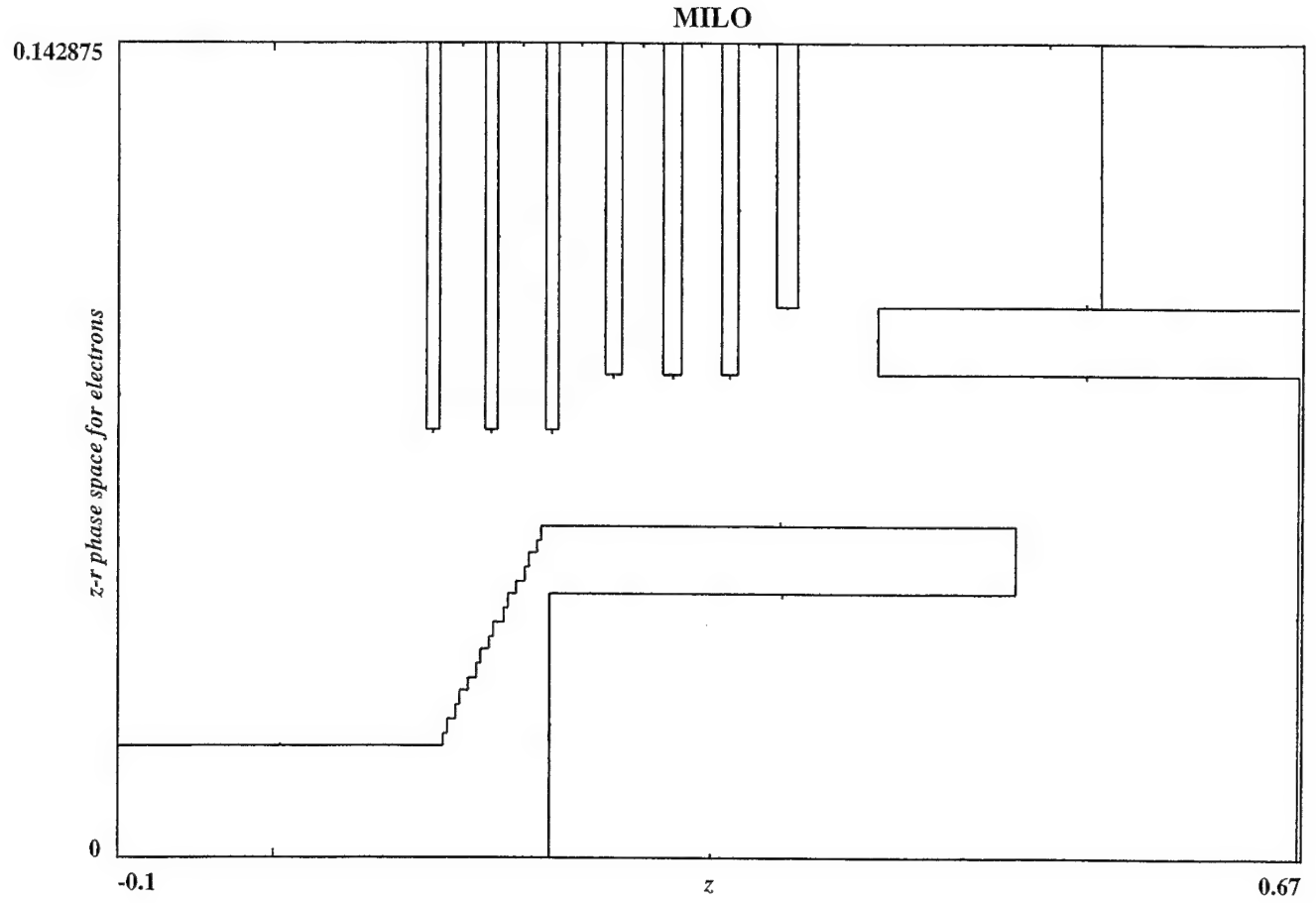
Error Flow Schematics



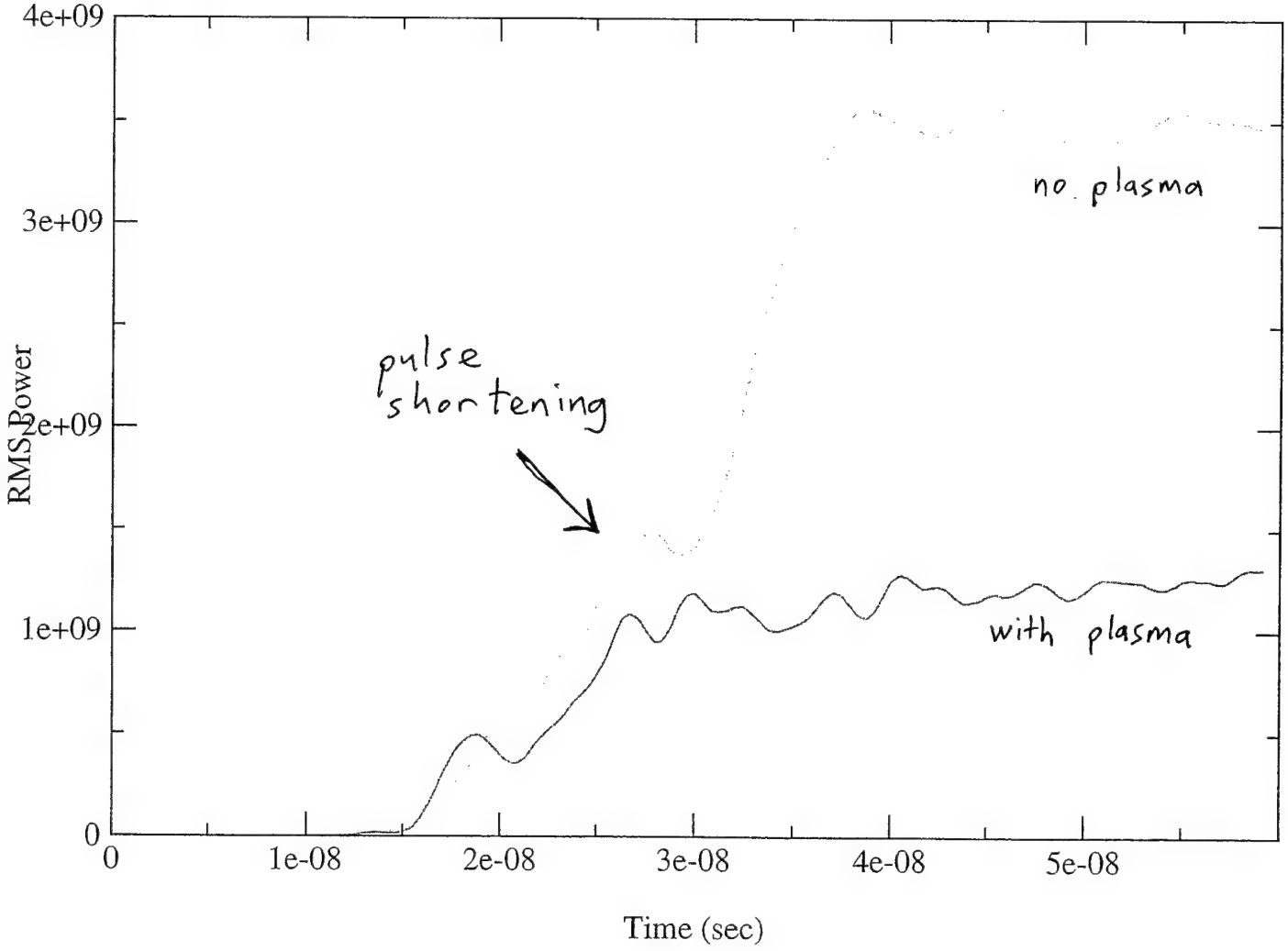
Using XOOPIC to Study Pulse Shortening in a MILO

- Keith Cartwright of AFRL (Kirtland AFB) is using XOOPIC to model pulse shortening in HPM devices
  - formation of plasma in the device can shorten the pulse
  - XOOPIC has two relevant models for this work
    - » ad hoc model for a plasma source
    - » detailed Monte Carlo collision package
  - the MC package can model complete physics
    - » ionization of the background gas by primary electrons
    - » full PIC treatment of resulting secondary electrons
      - including further ionization by secondaries
- We present some of Keith's XOOPIC plots
  - these are preliminary results, used with permission
  - any conceptual errors are mine alone

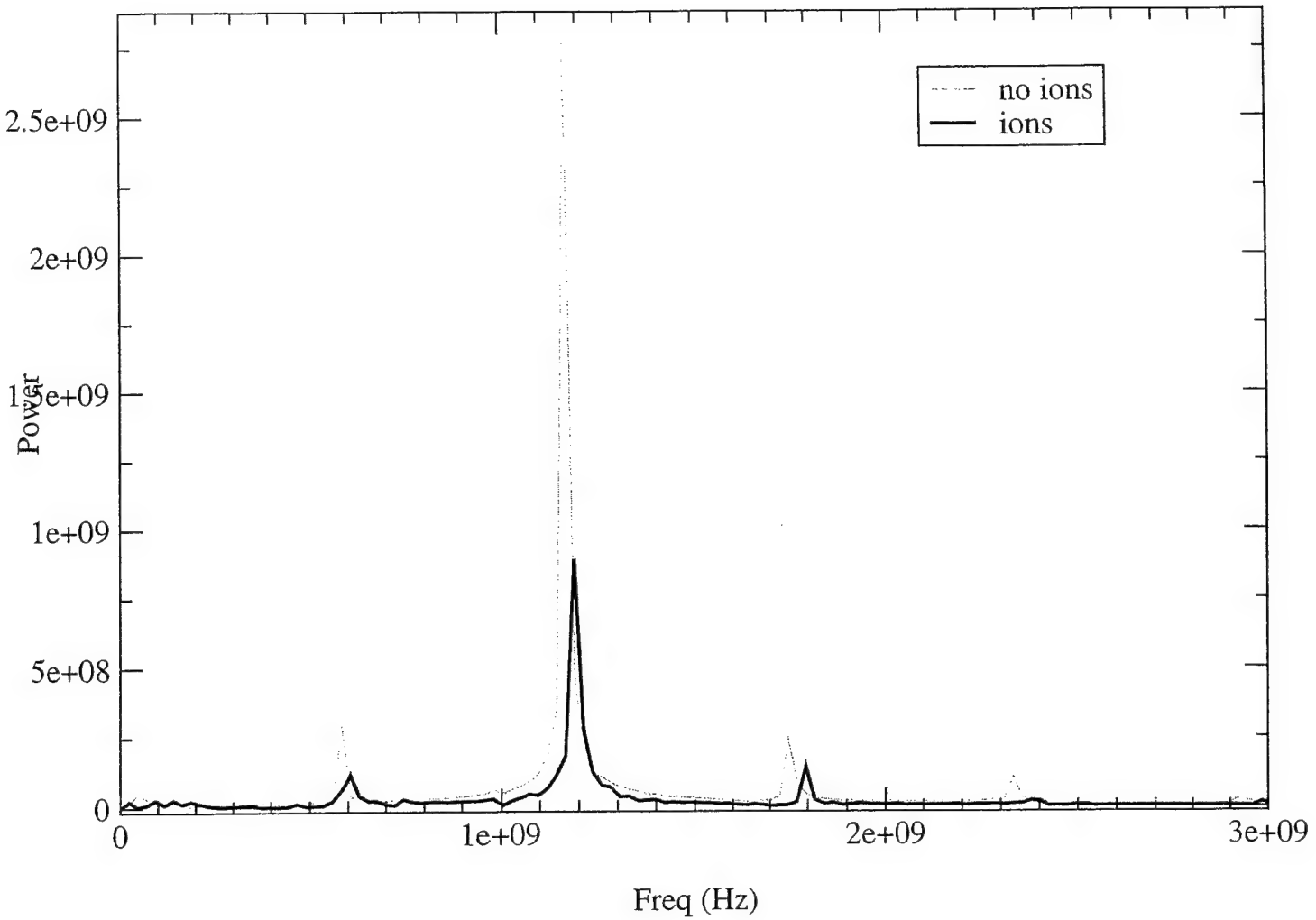
MILO -- as modeled in r-z geometry by XOOPIC



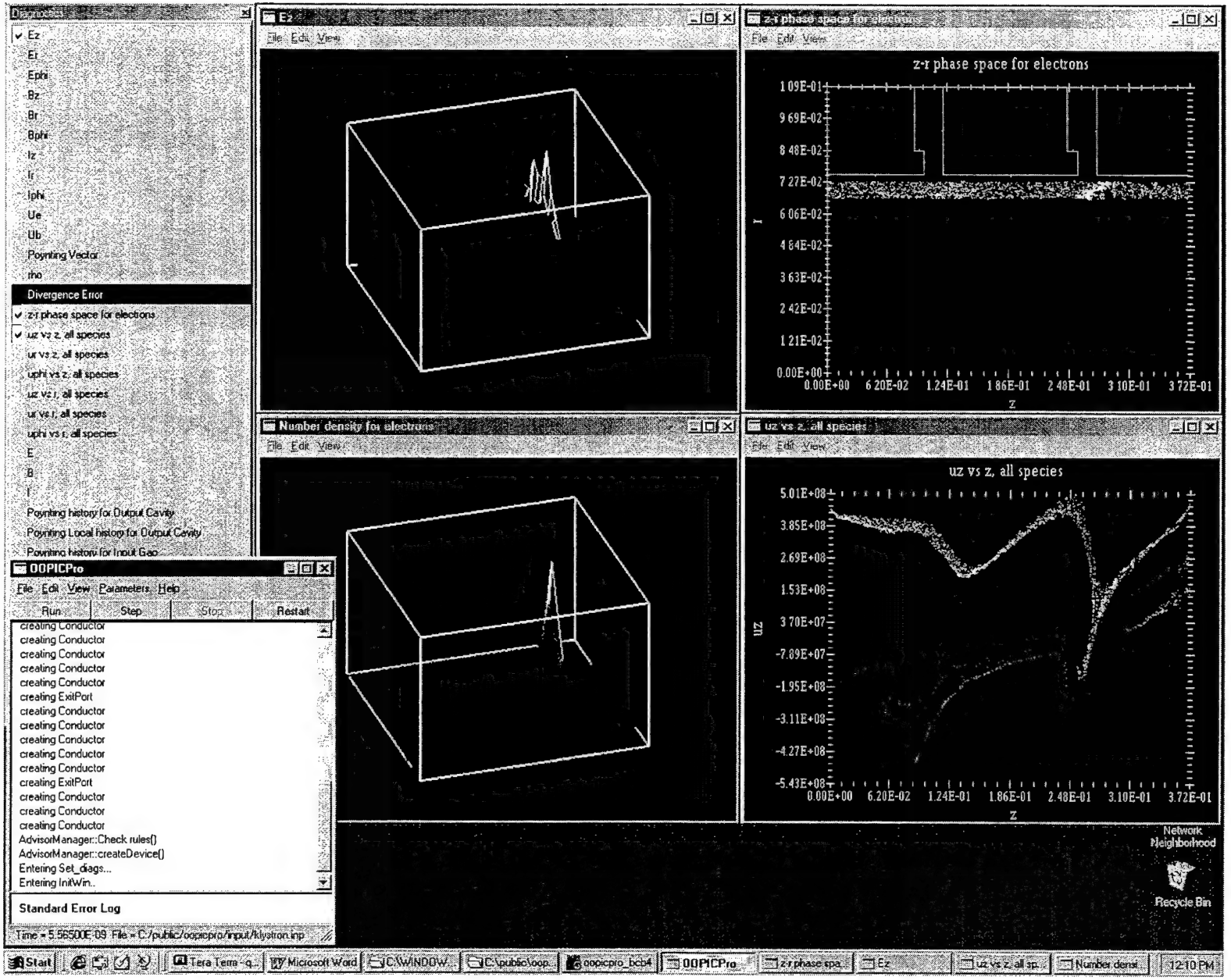
Plasma in MILO Reduces RMS Power



FFT Shows Power Reduction in the Primary Mode



OOPIC Pro -- a simple klystron simulation



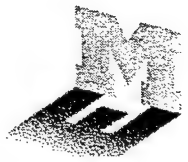
MichiganEngineering

Fluctuation Reduction Algorithms For Particle Simulations of Plasmas

Iain D. Boyd and Jing Fan
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI 48109

Funded by the Air Force Office of Scientific Research
through Tech-X Corporation

Overview



MichiganEngineering

- Background and motivation.
- Simulation methods:
  - Direct Simulation Monte Carlo (DSMC);
  - Information Preservation (IP);
  - Particle In Cell (PIC).
- Investigations:
  - unperturbed stream;
  - perturbed stream (Landau damping);
  - two-stream instability.
- Conclusions and future work.

Background and Motivation

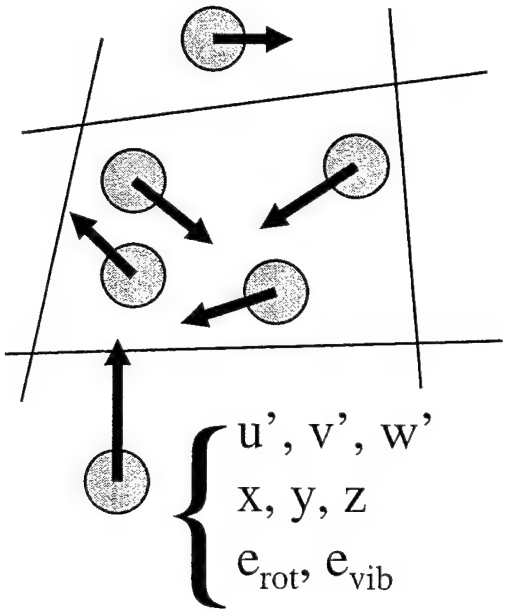
MichiganEngineering

- Particle methods suffer from statistical fluctuations:
  - use large particle count and/or large sample size.
- Direct simulation Monte Carlo method (DSMC):
  - particle method for rarefied gas flows;
  - fluctuation problems for low-speed MEMS flows;
  - Information Preservation method (IP) developed.
- Present investigation:
  - try to develop PIC-IP algorithm;
  - apply to standard, 1d, electro-static flows;
  - compare to results from the UC-XES1 code.

Particle Method for Gas Flows

MichiganEngineering

- Direct simulation Monte Carlo (DSMC):
  - developed by Bird (1960's);
  - particles move/collide in physical space;
  - particles possess microscopic properties, e.g. u' (thermal velocity);
  - cell size ~ λ;
  - time step ~ $1/\nu$;
  - collisions handled statistically;
  - ideal for supersonic/hypersonic flows;
  - large statistical scatter for subsonic flow.



Statistical Scatter in DSMC

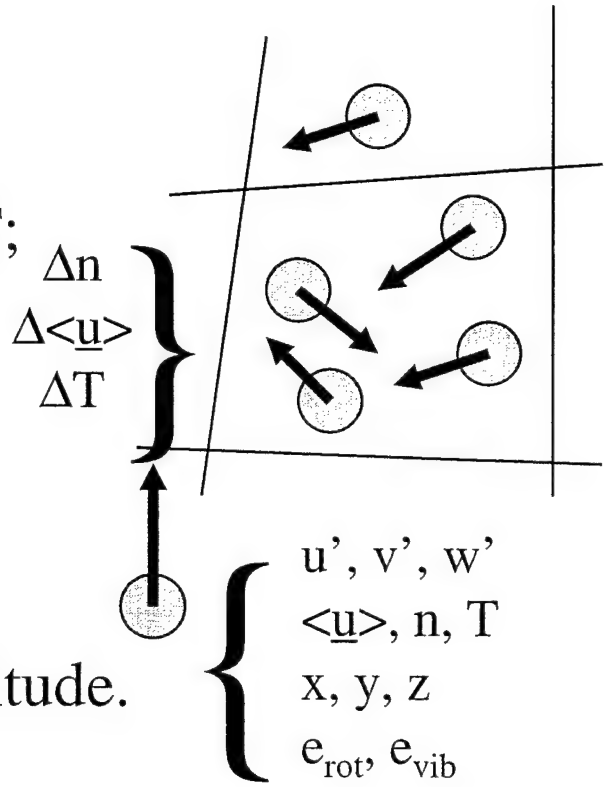
MichiganEngineering

- Typical speed in micro-scale flow V ~ 10 m/s.
- For air in equilibrium at room temperature:
  - physical standard deviation of velocity
    $$\sigma = \bullet 2RT = 400 \text{ m/s};$$
  - assuming Poisson process, statistical scatter in DSMC result (=noise) based on N particles
    $$\sigma' = \sigma / \bullet N$$
- For signal-to-noise ratio (V/σ') of 3:
  - $N = (3*400/10)^2 = 14{,}400$ particles (in each cell);
  - typically, N=20 giving (V/σ') = 0.11 !

Novel Particle Method

MichiganEngineering

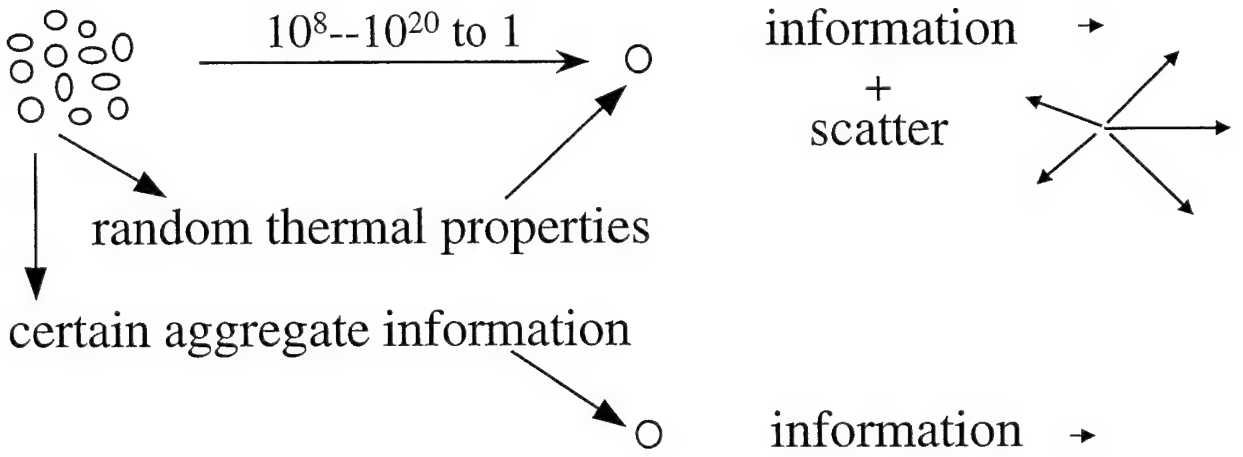
- Information Preservation (IP):
  - currently under development;
  - evolves alongside DSMC;
  - particles and cells possess preserved macroscopic information, e.g. n, $\langle \underline{u} \rangle$, T;
  - Δn from mass conservation;
  - $\Delta\langle \underline{u} \rangle$ from momentum conservation;
  - ΔT from energy conservation;
  - greatly reduces statistical fluctuations;
  - can reduce CPU time by order of magnitude.



Basic Idea of the IP Method

MichiganEngineering

real molecules simulated particles in DSMC



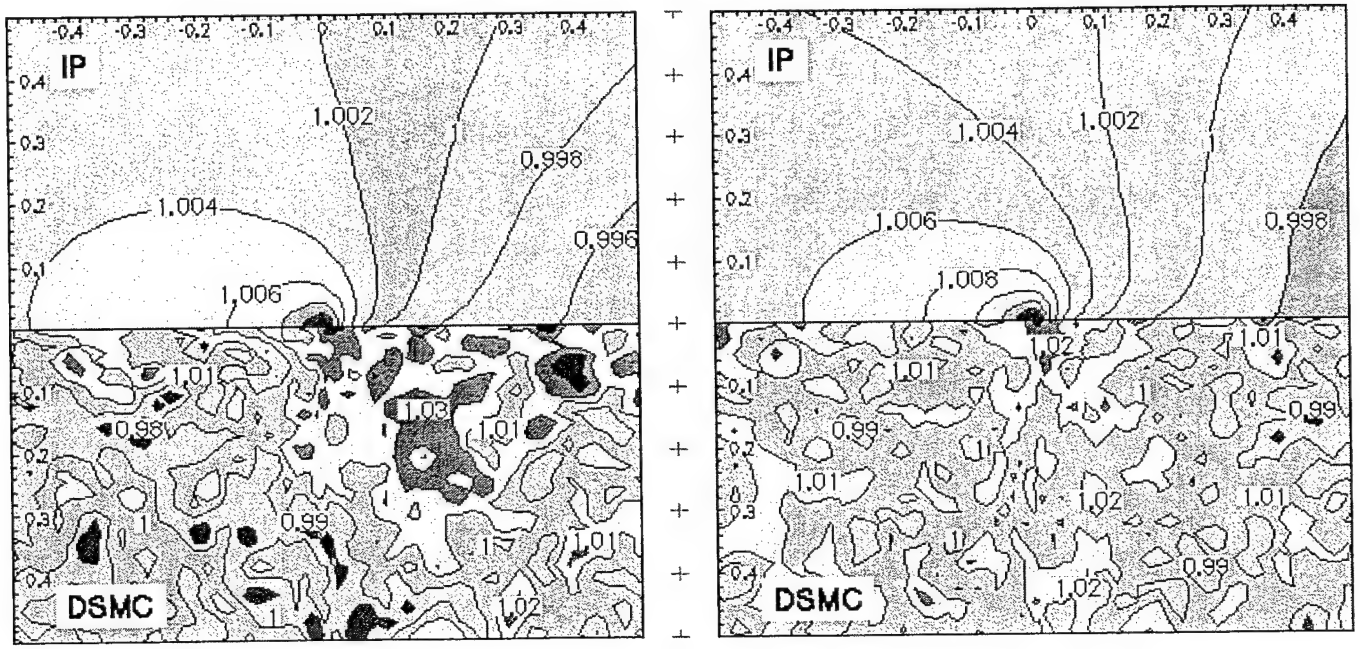
IP method

- preserve macroscopic information for each DSMC particle

Flow over a Flat Plate

MichiganEngineering

$Ma_\infty = 0.1$, $Kn_\infty = 0.01$ Total sample size:760,000 particles/cell



Density field

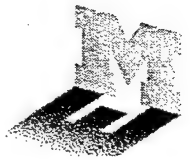
Pressure field

PIC-IP Algorithm

MichiganEngineering

- Incorporate IP algorithm into UC-XES1 code.
- Microscopic information for each particle:
  - position;
  - velocity (stream + thermal).
- Macroscopic information for each cell and particle:
  - density (continuity equation);
  - velocity (momentum equation including self-consistent electro-static body force);
  - temperature (particle microscopic velocities).

Results for Warm Plasmas



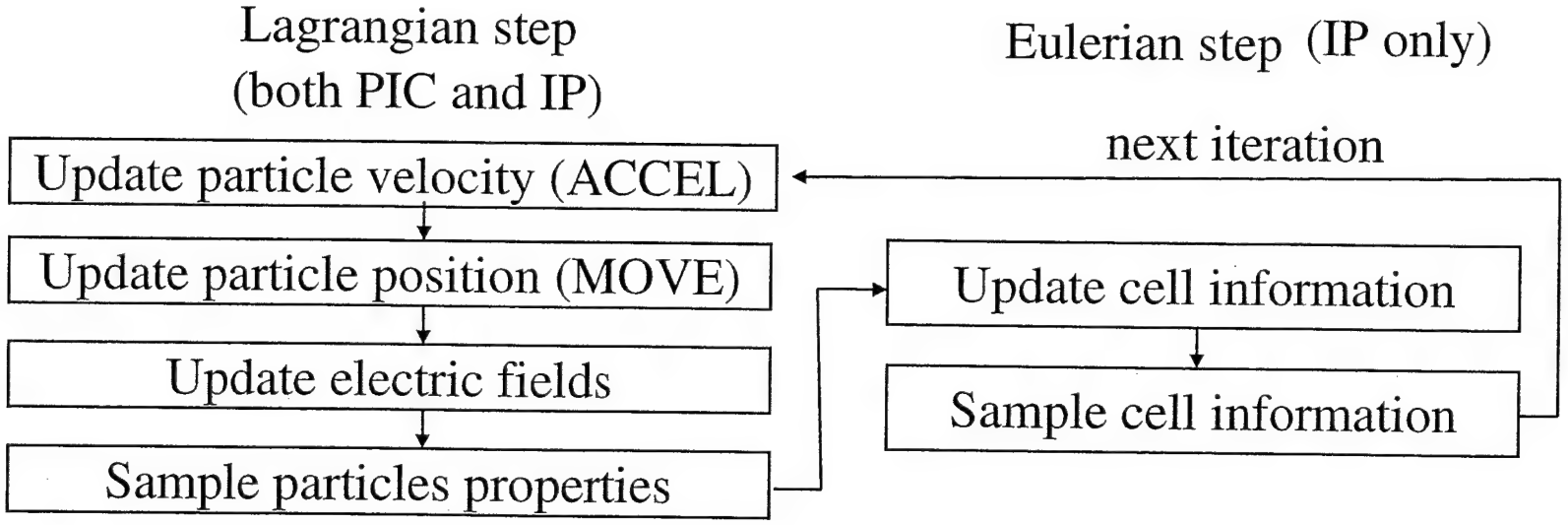
MichiganEngineering

- Unperturbed stream:
  - v_1=3, v_t=0.5, L=2π;
  - Δt=0.2, Ng=32, it=10,000, Nc=50-5000.
- Perturbed stream (landau damping):
  - v_1=0, initial perturbation: $\delta x = 0.1 \cos(2\pi x/L)$.
- Two-stream instability:
  - v_1=±1, L=10π;
  - initial perturbation: $\delta x = \pm 0.0001 \cos(2\pi x/L)$;
  - Δt=0.1, Ng=128, it=300, Nc=64-4096.

Algorithm of the IP Method

MichiganEngineering

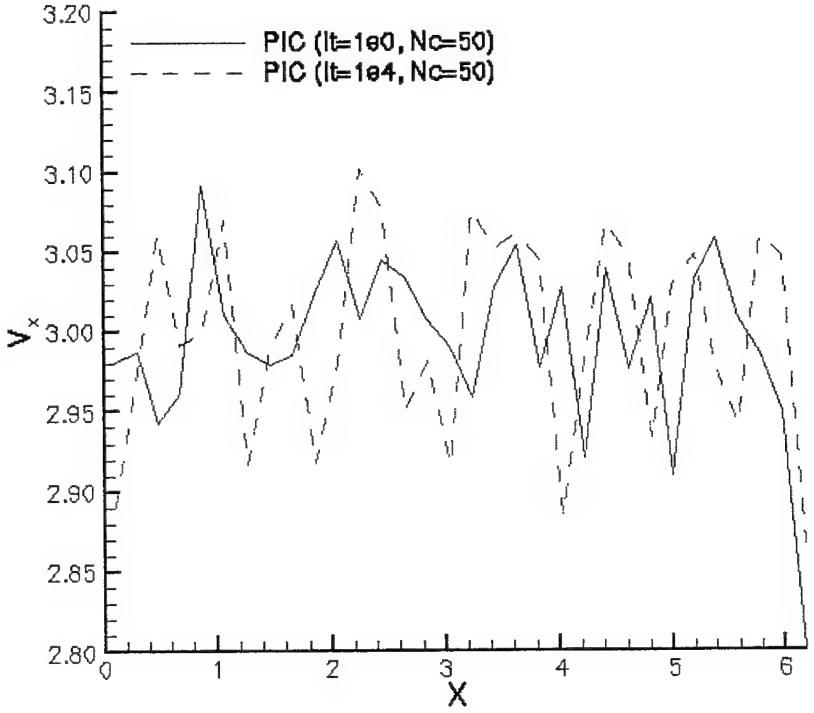
- The IP method:
  - usual microscopic movement in PIC (Lagrangian frame);
  - update the macroscopic information according to conservation laws (Eulerian frame).
- There are two steps at each iteration:

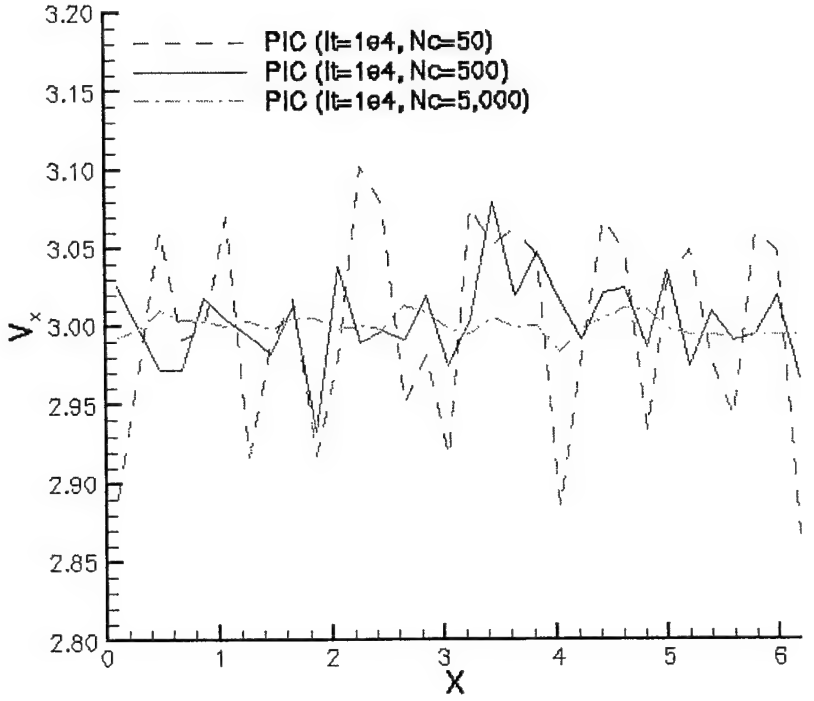


Unperturbed Stream

MichiganEngineering

Velocity Profiles

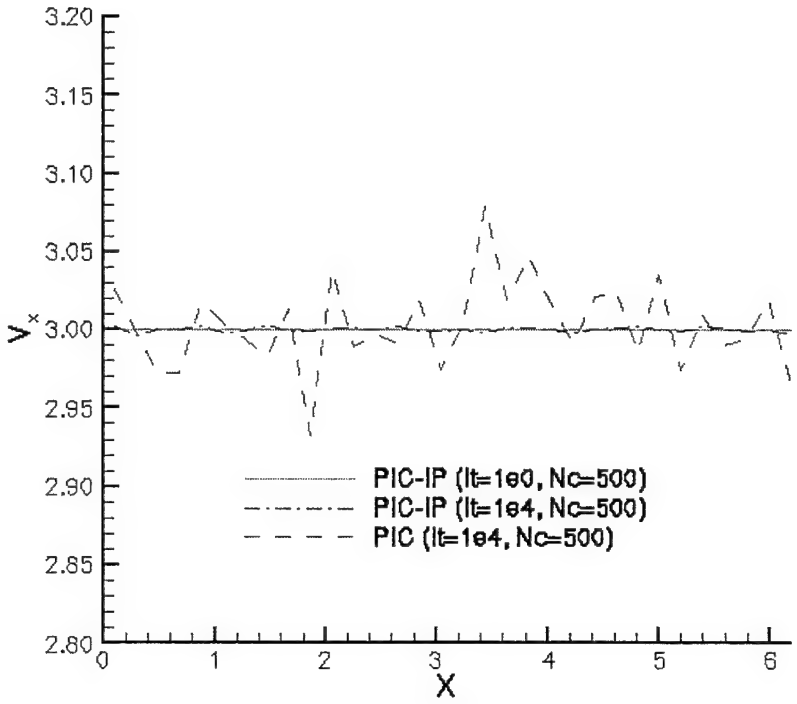


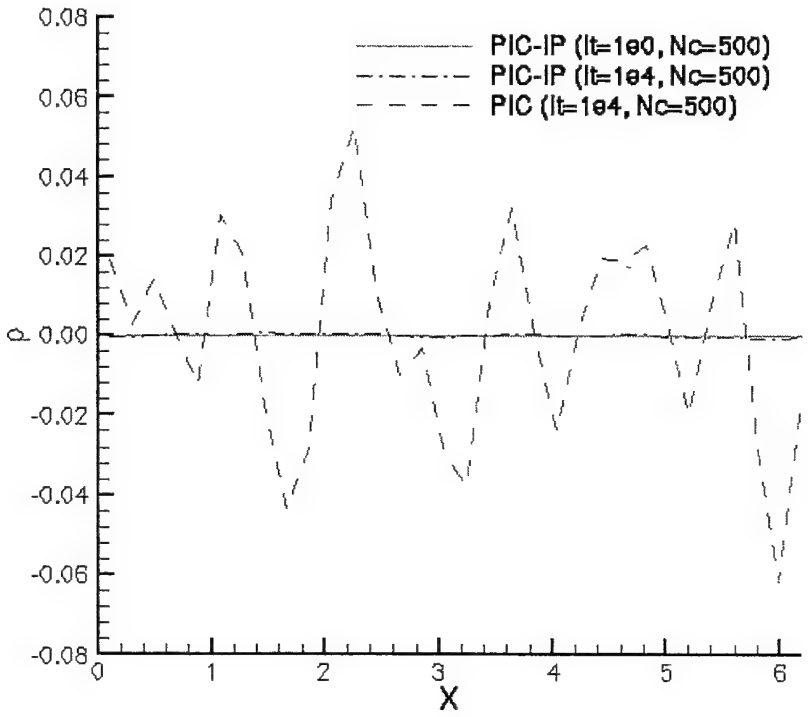


Unperturbed Stream

MichiganEngineering

Comparison of PIC and IC-IP Profiles





Numerical Analysis

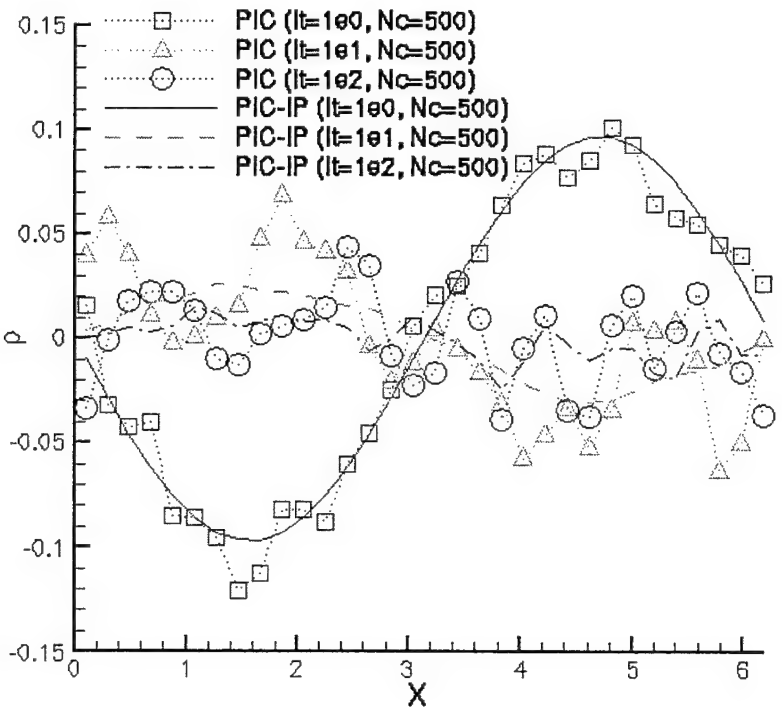
MichiganEngineering

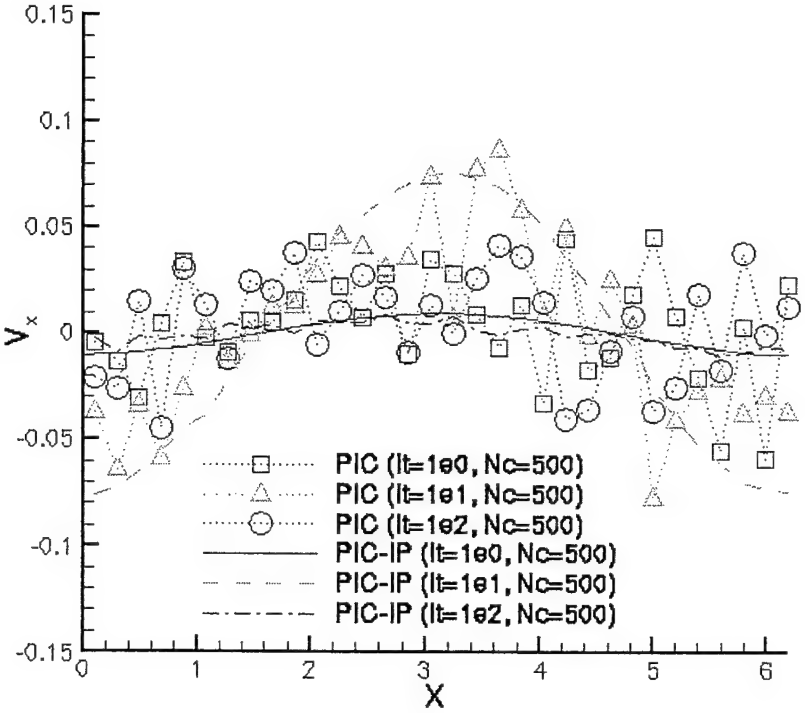
Method	Particles Per Cell	Iteration	σ	CPU (sec)
PIC	50	10,000	0.065	25
PIC	500	10,000	0.016	103
PIC	5,000	10,000	0.006	915
PIC (quiet)	50	1	0.022	0.3
PIC (quiet)	50	10,000	0.079	25
PIC-IP	500	10,000	0.0054	115

Perturbed Stream

MichiganEngineering

Landau Damping

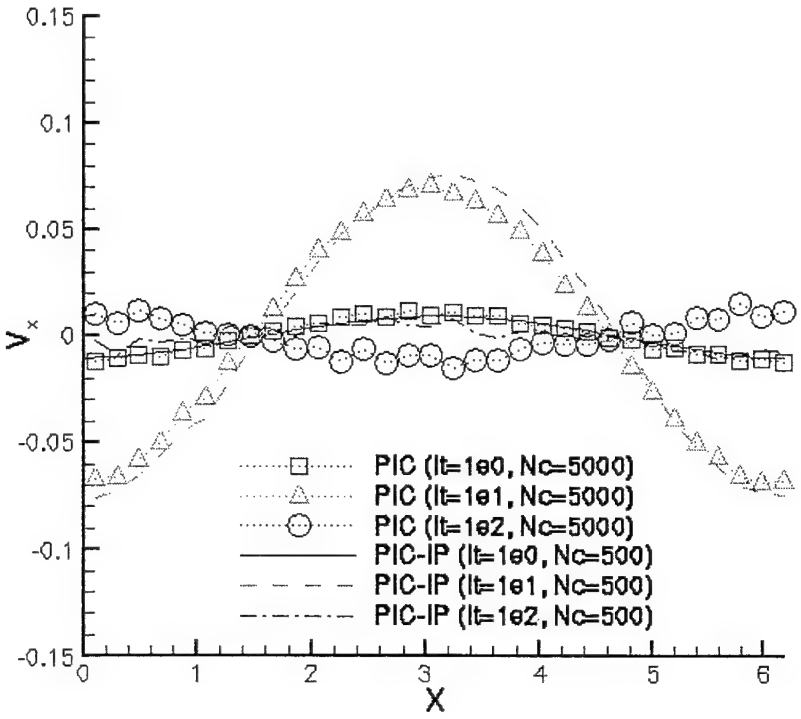


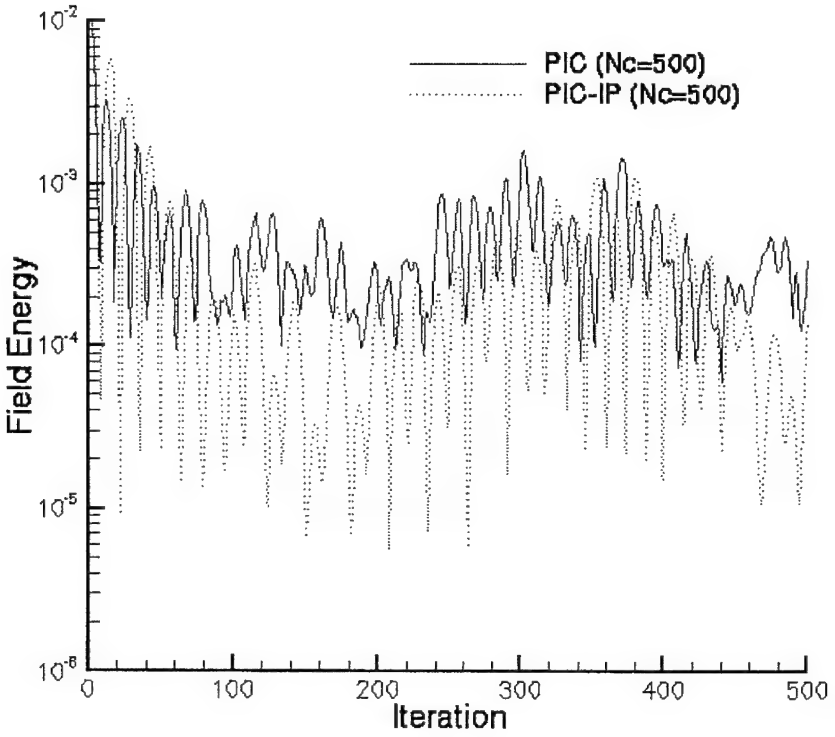


Perturbed Stream

MichiganEngineering

Landau Damping

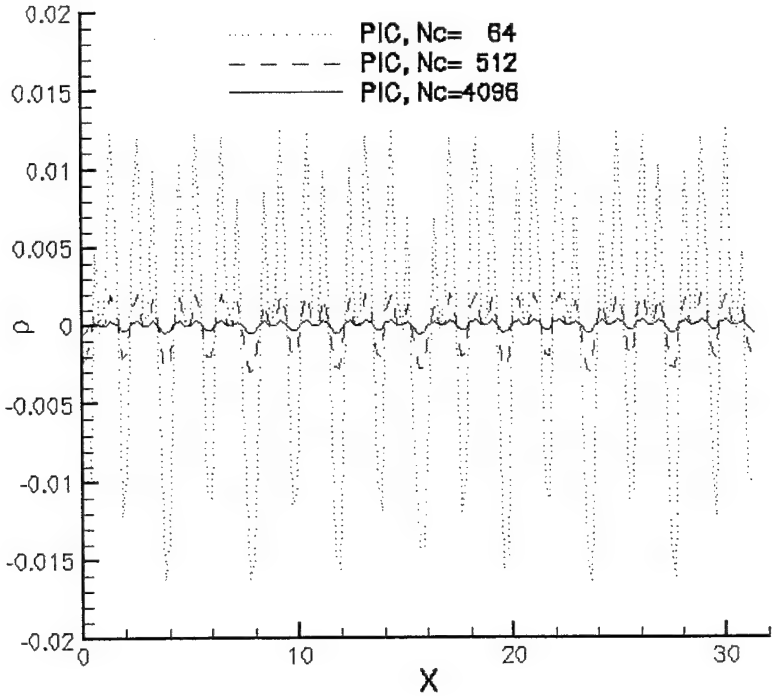




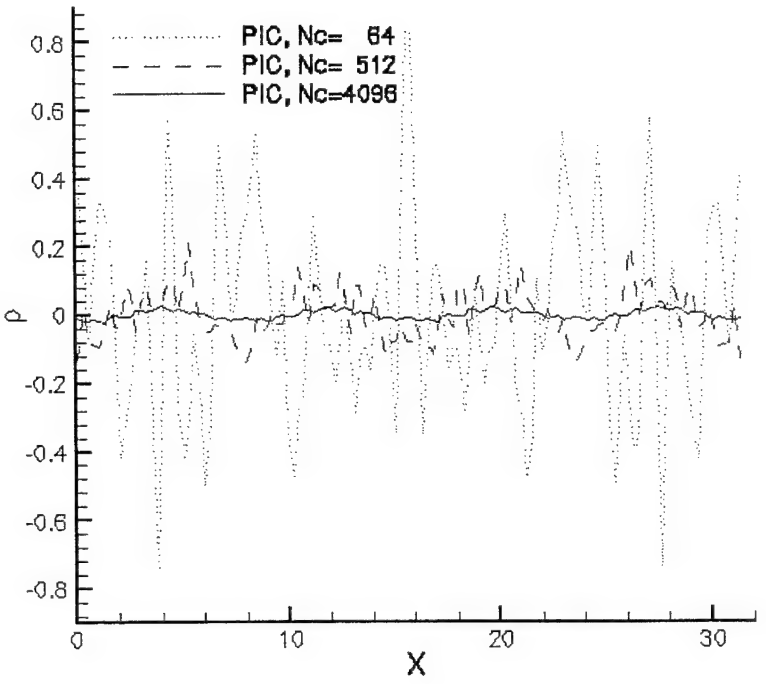
Two-Stream Instability

MichiganEngineering

Initial Profiles



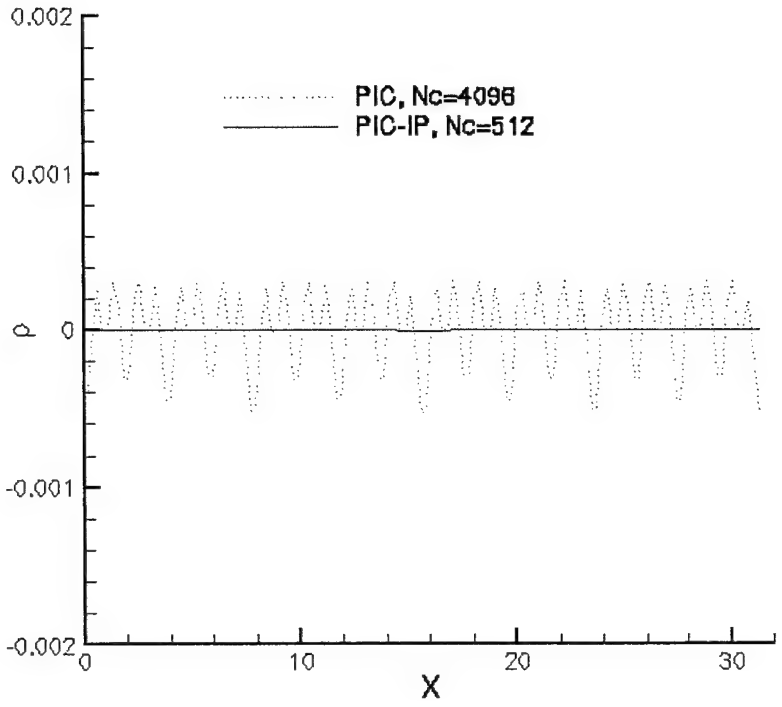
Final Profiles



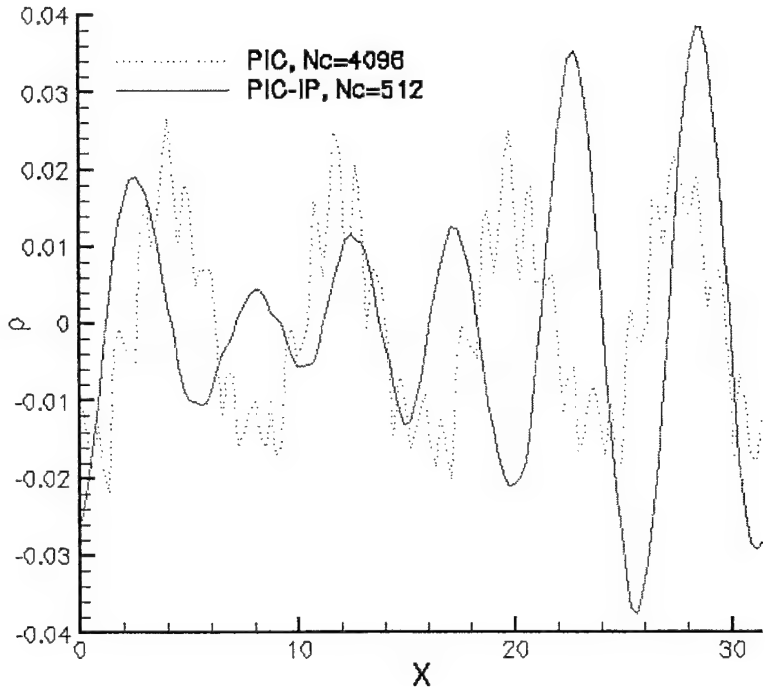
Two-Stream Instability

MichiganEngineering

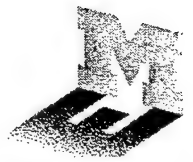
Initial Profiles



Final Profiles



Conclusions



MichiganEngineering

- PIC simulations of warm plasmas:
  - significant fluctuations with 50 particles/cell;
  - fluctuations reduced by factor of 10 by increasing total number of particles by factor of 100;
  - for instability, final results depend on number of particles/cell.
- PIC-IP simulations of warm plasmas:
  - algorithm successfully implemented in XES1 code;
  - reduces fluctuations while maintainingphysical accuracy;
  - factors of 6-8 reduction in CPU time for similar results and levels of statistical fluctuation.

Future Work

MichiganEngineering

- Further study of present PIC-IP algorithm:
  - effect of other fluctuation reduction schemes (e.g. quiet start);
  - cold plasma conditions;
  - verification of method against data.
- Extensions of present PIC-IP algorithm:
  - electron energy equation for temperature;
  - effects of applied EM fields;
  - multi-dimensionality.

Accuracy Analysis of the PIC Method

J. P. Verboncoeur and K. L. Cartwright
Dept. Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720-1770

Acknowledgments: This work was supported by AFOSR STTR contract F49620-99-C-0028.

The discretization errors for many steps of the classical Particle-in-Cell (PIC) model have been well-studied [1][2]. In this work, the errors in the interpolation algorithms, which provide the connection between continuum particles and discrete fields, are described in greater detail. In addition, the coupling of errors between steps in the method is discussed. The analysis is carried out for both electrostatic and electromagnetic PIC models.

Outline

1. Electrostatic PIC error
   (a) Truncation error in each step
      i. interpolate x$\rightarrow \rho$
      ii. Poisson Equation
      iii. E from ϕ
      iv. interpolate E to x (and static B)
      v. Lorentz Equations
   (b) coupled error
2. Electromagnetic PIC error
   (a) Truncation error in each step.
      i. interpolate x,v$\rightarrow$J
      ii. Ampere
      iii. Faraday
      iv. B(Yee) to B(Mesh) and E(Yee) to E(Mesh)
      v. Interpolate B$\rightarrow$x and E$\rightarrow$x
      vi. Lorentz Equations
   (b) coupled error

Analysis of Errors Inherent in the Generic PIC Algorithm

The error in a generic PIC algorithm have been analyzed separately in the infinite electrostatic limit by Birdsall and Langdon [1] and Hockney and Eastwood [2]. Our methodology differs from previous method in:

- We will use a method to analyze the error that can take into account boundaries. (Will not assume an infinite space and not use spectral methods.)
- We will retain spacial and temporal discretization errors (Δx, Δt) as well macroparticle dependence.

Electrostatic Klimontovich Formulation [3]

The density of N_s of species s is:

$$N_s(\mathbf{x}, \mathbf{v}, t) = \sum_{i=1}^{N_0} \delta(\mathbf{x} - \mathbf{X}_i(t))\delta(\mathbf{v} - \mathbf{V}_i(t)) \quad (1)$$

The microscopic charge density is:

$$\rho(\mathbf{x}, t) = \sum_s q_s \int N_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} = \sum_s q_s \sum_{i=1}^{N_0} \delta(\mathbf{x} - \mathbf{X}_i(t)) \quad (2)$$

Poisson Equation

$$\nabla^2 \phi(\mathbf{x}, t) = -\frac{\rho(\mathbf{x}, t)}{\epsilon_0} \text{ with } \mathbf{E}(\mathbf{x}, t) = -\nabla\phi(\mathbf{x}, t) \quad (3)$$

Lorentz equation

$$m_s \ddot{\mathbf{X}}_i(t) = q_s(\mathbf{E}(\mathbf{X_i}, t) + \mathbf{V}_i(t) \times \mathbf{B}(\mathbf{X_i}, t) \quad (4)$$

Electrostatic PIC Formulation

The charge density is now defined on a grid at time level n.

$$\rho_{\mathbf{j}}(n) = \sum_s q_s \int N_s(\mathbf{x}, \mathbf{v}, n) d\mathbf{v} = \sum_s q_s \sum_{\mathbf{j}=0}^{\mathbf{J}} \sum_{i=1}^{N_p} W_1(\mathbf{x_j} - \mathbf{X}_i(n)) \tag{5}$$

Where W is the weighting function.

Poisson Equation written with a general finite difference operator, D_x:

$$D_x^2 \phi_{\mathbf{j}}(n) = -\frac{\rho_{\mathbf{j}}(n)}{\epsilon_0} \text{ with } \mathbf{E_j}(n) = -D_x \phi_{\mathbf{j}}(n) \tag{6}$$

Lorentz equation written with a general finite difference operator, D_t:

$$m_s D_t^2 \mathbf{X}_i(n) = q_s(\mathbf{E}_i + D_t \mathbf{X}_i(n) \times \mathbf{B}_i) \tag{7}$$

with

$$\mathbf{E}_i = \sum_{\mathbf{j}} W_2(\mathbf{x_j} - \mathbf{X}_i(n)) \mathbf{E}_{\mathbf{x_j}} \text{ similarly } \mathbf{B}_i = \sum_{\mathbf{j}} W_3(\mathbf{x_j} - \mathbf{X}_i(n)) \mathbf{B}_{\mathbf{x_j}} \tag{8}$$

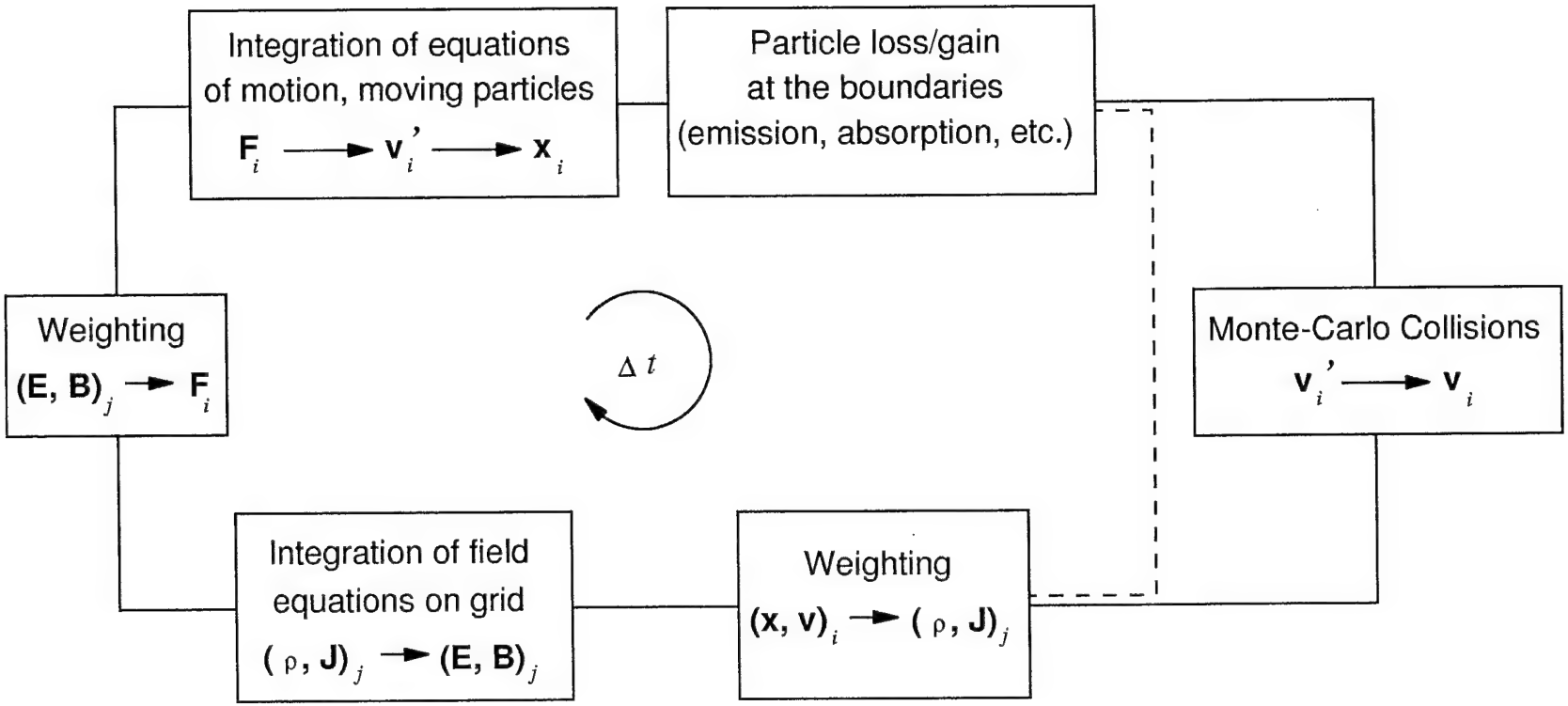


Figure 1: PIC flow schematic.

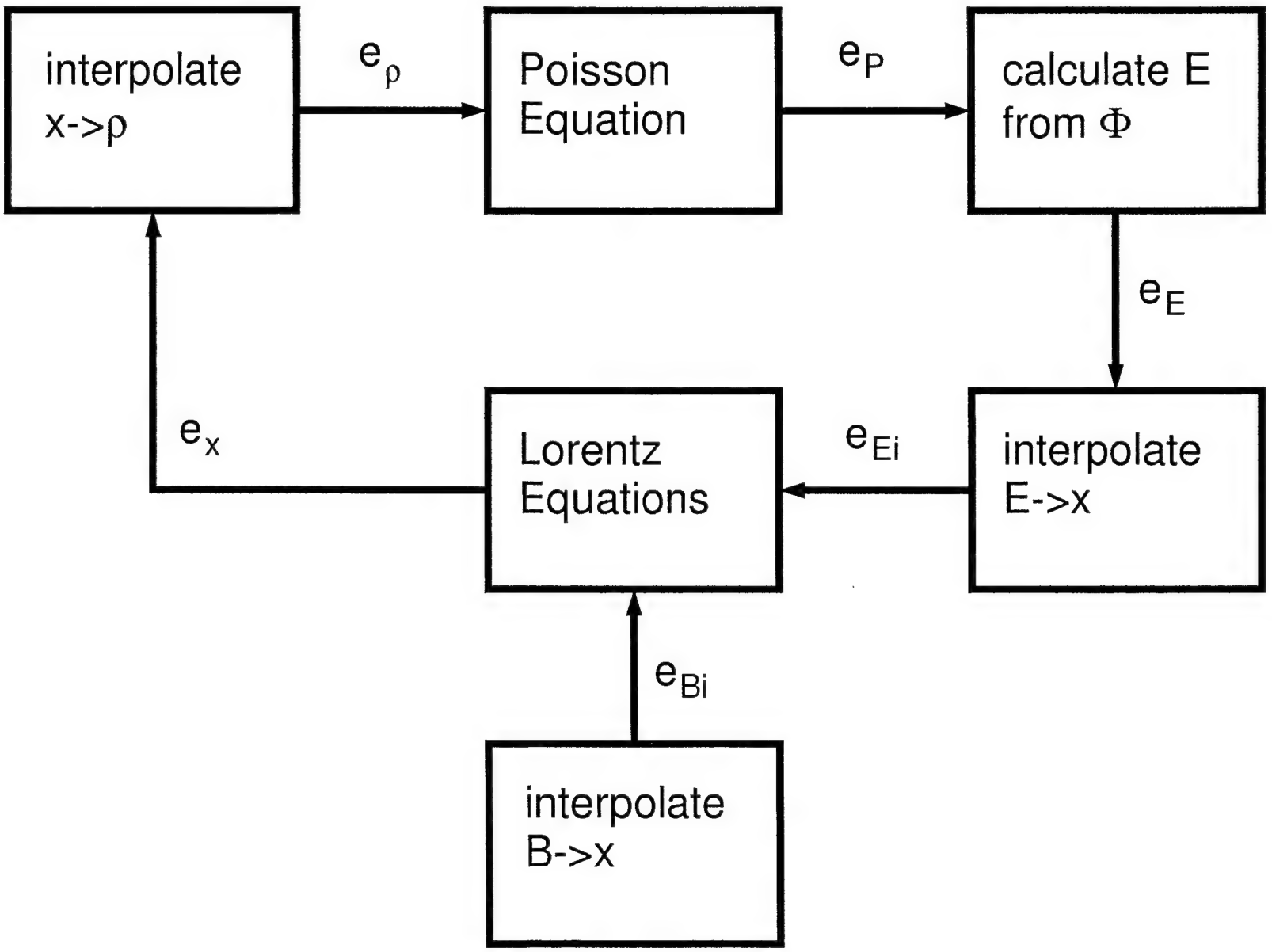


Figure 2: Electrostatic PIC error flow schematic.

It is a straight forward but tedious process to calculate the truncation error for equations 5 - 8. The results for various weighting scheme (NGP, linear, and quadratic) have been obtained, but only linear will be shown here with a Leap-Frog integrator, center difference Poisson Equation in 1d, and no magnetic field. This will give us the error in one update. With this formalism in place we can analyze effectiveness of weighting functions (smoothing and sub-cycling can also be added). The effect of using fewer finite shaped particle computer particles versus delta function particles also needs further analysis.

Linear Weightings

Linear weighting of ρ:

$$e_{\rho,lin} \equiv \rho_j - \rho(x_j) = \frac{\Delta x^2}{12}\rho^{(2)}(x_j) + \frac{\Delta x^4}{360}\rho^{(4)}(x_j) + \mathcal{O}(\Delta x^6) \quad (9)$$

Poisson Error

$$e_P \equiv \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2} - \frac{\partial^2\phi(x_j)}{\partial x^2} = \frac{\Delta x^2}{12}\phi^{(4)}(x_j) + \frac{\Delta x^4}{360}\phi^{(6)}(x_j) + \mathcal{O}(\Delta x^8) \quad (10)$$

Note: these two errors cancel to all orders ($\frac{\partial^2\phi(x)}{\partial x^2} = -\frac{\rho(x)}{\epsilon_0}$). Field Interpolation Error

$$e_{field,lin} \equiv (1-f)E_j + fE_{j+1} - E(x_i) = \frac{3f^2 - 3f - 1}{6}\Delta x^2\phi^{(3)}(x_i) + \mathcal{O}(\Delta x^3) \quad (11)$$

where $x_i = x_j + f\Delta x$.

Lorentz Equation

$$e_a \equiv \frac{X_i(n+1) - 2X_i(n) + X_i(n-1)}{\Delta t^2} - \ddot{X}(t_n) = \frac{\Delta t^2}{12}X_i^{(4)}(t_n) + \frac{\Delta t^4}{360}X_i^{(6)}(t_n) + \mathcal{O}(\Delta t^8) \quad (12)$$

The total error from this loop is (Lorentz plus Field Interpolation because of the cancellation of Weighting and Poisson Error):

$$\begin{aligned} e_{total} &= m_s \frac{\Delta t^2}{12} X_i^{(4)}(t_n) + q_s \frac{3f^2 - 3f - 1}{6} \Delta x^2 \phi^{(3)}(x_i, t_n) \\ &= q_s \left(\frac{\Delta t^2}{12} \phi^{(1,2)}(x_i, t_n) - \frac{3f^2 - 3f - 1}{6} \Delta x^2 \phi^{(3)}(x_i, t_n) \right) \end{aligned} \quad (13)$$

From a simulation $\phi^{(1,2)}(x_i, t_n)$ and $\phi^{(3)}(x_i, t_n)$ can be calculated and a more optimal Δt and Δx can be found.

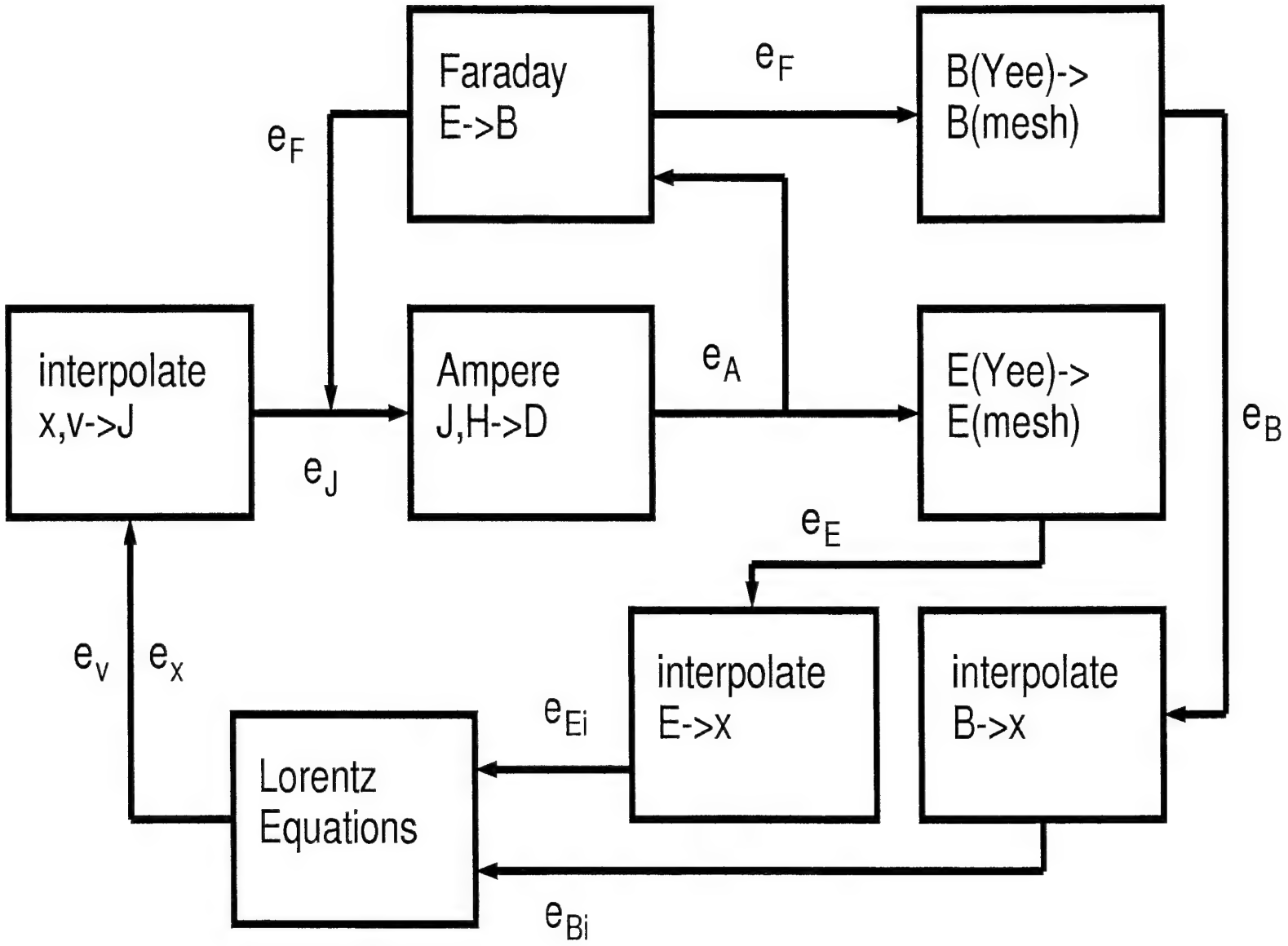


Figure 3: Electromagnetic PIC error flow schematic.

Electromagnetic Klimontovich Formulation

In addition to the electrostatic equations.

The microscopic current is:

$$J(\mathbf{x},t) = \sum_s q_s \int N_s(\mathbf{x},\mathbf{v},t)\mathbf{v}d\mathbf{v} = \sum_s q_s \sum_{i=1}^{N_0} \mathbf{V}_i(t)\delta(\mathbf{x}-\mathbf{X}_i(t)) \tag{14}$$

Faraday's equation:

$$\nabla \times \mathbf{E}(\mathbf{x},t) = -\frac{\partial \mathbf{B}(\mathbf{x},t)}{\partial t} \tag{15}$$

Ampere's equation:

$$\nabla \times \mathbf{B}(\mathbf{x},t) = \mu_0 J(\mathbf{x},t) + \mu_0\epsilon_0 \frac{\partial \mathbf{E}(\mathbf{x},t)}{\partial t} \tag{16}$$

Relativistic Neutron-Lorentz equation

$$m_s \frac{\partial(\gamma \dot{\mathbf{X}}_i(t))}{\partial t} = q_s(\mathbf{E}(\mathbf{X_i},t) + \mathbf{V}_i(t) \times \mathbf{B}(\mathbf{X_i},t) \tag{17}$$

Electromagnetic PIC Formulation

In addition to electrostatic equations.

Current weighting:

$$\mathbf{J_j}(n+\frac{1}{2}) = \sum_s q_s \sum_{\mathbf{j}=0}^{\mathbf{J}} \sum_{i=1}^{N_p} \mathbf{V}_i(n+\frac{1}{2}) W(\mathbf{x_j} - (\mathbf{X}_i(n) + \mathbf{X}_i(n+1))/2) \qquad (18)$$

The weighting function for each direction can be different.

Symmetric current weightings are not charge conserving (zero order in Poisson Equation) NGP interpolation in the direction of the current component and linear interpolation in the transverse direction is.

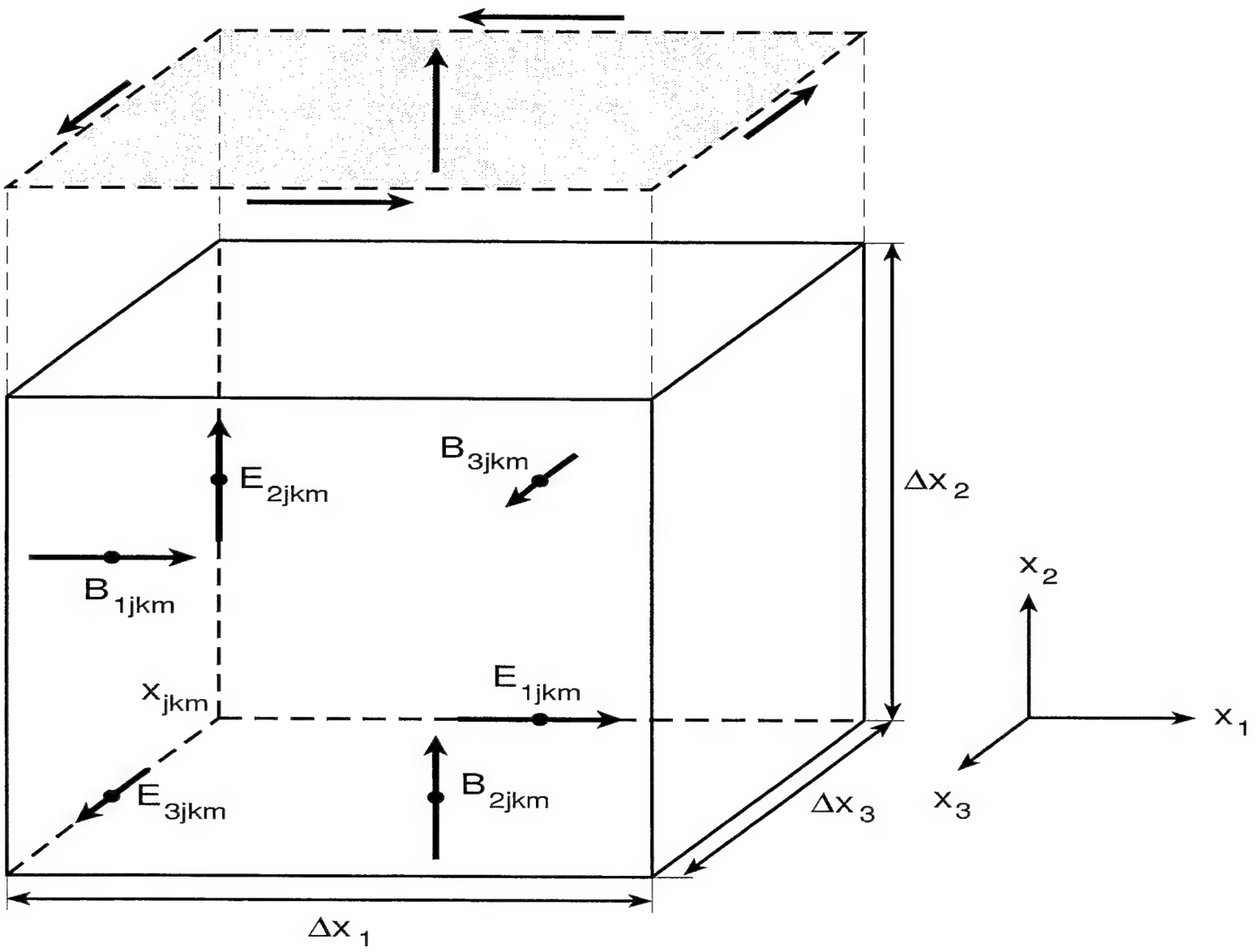


Figure 4: Yee mesh.

Solving Ampere's and Faraday's equation on a Yee mesh

Defining line and surface integrals:

$$\mathcal{E}_{\alpha,j,k,m} = \int_{\text{edge}} \mathbf{E} \cdot d\mathbf{l} = E_{\alpha,j,k,m} \Delta x_{\alpha,\beta+1/2} \tag{19}$$

$$\mathcal{H}_{\alpha,j,k,m} = \int_{\text{edge}} \mathbf{H} \cdot d\mathbf{l} = B_{\alpha,j,k,m} \Delta x_{\alpha,\beta} / \mu_{j,k,m} \tag{20}$$

$$\mathcal{D}_{\alpha,j,k,m} = \int_{\text{face}} \mathbf{D} \cdot d\mathbf{S} = \varepsilon_{j,k,m} E_{\alpha,j,k,m} \Delta x_{\alpha',\beta+1/2} \Delta x_{\alpha'',\beta+1/2} \tag{21}$$

$$\mathcal{B}_{\alpha,j,k,m} = \int_{\text{face}} \mathbf{B} \cdot d\mathbf{S} = B_{\alpha,j,k,m} \Delta x_{\alpha',\beta} \Delta x_{\alpha'',\beta} \tag{22}$$

Here $\alpha' \equiv \text{mod}_3(\alpha+1)$, $\alpha'' \equiv \text{mod}_3(\alpha+2)$ are right-handed permutations of the index α.

The finite difference forms of the macroscopic Maxwell Equations become:

$$\frac{\mathcal{D}^{t+\Delta t}_{\alpha,j,k,m} - \mathcal{D}^{t}_{\alpha,j,k,m}}{\Delta t} = -\mathcal{H}^{t+\Delta t/2}_{\alpha',j,k,m} + \mathcal{H}^{t+\Delta t/2}_{\alpha'(,j,k,m)-1} \tag{23}$$

$$+\mathcal{H}^{t+\Delta t/2}_{\alpha'',j,k,m} - \mathcal{H}^{t+\Delta t/2}_{\alpha''(,j,k,m)-1} - I^{t+\Delta t/2}_{\alpha,j,k,m} \tag{24}$$

$$\frac{\mathcal{B}^{t+\Delta t/2}_{\alpha,j,k,m} - \mathcal{B}^{t-\Delta t/2}_{\alpha,j,k,m}}{\Delta t} = \mathcal{E}^{t}_{\alpha'(,j,k,m)+1} - \mathcal{E}^{t}_{\alpha',j,k,m} \tag{25}$$

$$-\mathcal{E}^{t}_{\alpha'',(j,k,m)+1} + \mathcal{E}^{t}_{\alpha'',j,k,m} \tag{26}$$

with the constitutive equations:

$$\mathcal{E} = \mathcal{C}^{-1}\mathcal{D} \tag{27}$$

$$\mathcal{H} = \mathcal{L}^{-1}\mathcal{B} \tag{28}$$

Here $\mathcal{C}$ and $\mathcal{L}$ have the units of capacitance and inductance, respectively, and store the geometric constants of the mesh and the permittivity and permeability.

- From the electromagnetic error flow schematic you can see that combining the error is much more complicated.
- Different directions couple together through γ in the Lorentz equation.
- The curl equations are at least 2d making the algebra much more tedious.
- The staggered field positions on the Yee mesh also complicates the error analysis.

- Symmetric current weightings are not charge conserving (zero order in Poisson Equation) NGP interpolation in the direction of the current component and linear interpolation in the transverse direction is.
- The relativistic Lorentz equation is still second order; however the leading order term is proportional to γ^6
- Because of the multiple branches a complete cycle through the PIC flow chart has not been done. The PIC (fixed fields) and Maxwell's equations have been done separately.
- The dispersion of EM plasma waves has not been done and is not in the literature. (Free space dispersion has been done for this field solve and many others.)

References

[1] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* Adam Hilger, New York, 1991.

[2] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles.* McGraw-Hill, New York, 1981.

[3] D. R. Nicholson. *Introduction to Plasma Theory.* Krieger Publishing Company, Malabar, Florida, 1992.

FINAL TECHNICAL REPORT

Fluctuation Reduction Algorithms For Particle Simulations of Plasma

Iain D. Boyd and Jing Fan
Department of Aerospace Engineering
University of Michigan
Ann Arbor, Michigan

1. Introduction

The Particle in Cell (PIC) technique is a widely used particle method for simulating plasma dynamics [1]. As with many particle techniques, the solutions provided by the PIC method can suffer from significant statistical fluctuations under certain physical conditions of interest. In principle, these fluctuations can be reduced by a combination of using an increased number of particles in a computation and by time-averaging the solution over a large number of iterations. Both of these procedures, however, lead to significant increases in the overall solution time and may not work for all conditions.

The direct simulation Monte Carlo (DSMC) is a widely used particle method for simulating neutral, rarefied gas flows [2]. The solutions generated by the DSMC technique also suffer from problems of statistical noise, particularly for low-speed flows that require a large number of samples to reduce the noise. Such a simulation is extremely time-consuming and beyond the capabilities of current computers [3]. To solve this problem, a new particle method called the Information Preservation (IP) algorithm was developed [4-6].

In the present report, we investigate whether the IP algorithm can be developed to reduce the statistical noise in PIC simulations. A hybrid PIC/IP algorithm is described first. Then the hybrid method is assessed in comparison to the standard PIC approach through its application to three standard flows: (1) an unperturbed plasma beam; (2) a perturbed plasma beam (landau damping); and (3) a two-stream instability problem. Finally, some conclusions and suggestions for further study are provided.

2. Combination of IP and PIC methods

The basic premise of the IP technique is to associate both microscopic and macroscopic information with each particle in a computer simulation [4-6]. The usual PIC microscopic steps are applied to the microscopic properties, and simple conservation laws are applied to the macroscopic properties. Macroscopic quantities are obtained through time or ensemble averaging of the information.

To clarify why the IP method works, compare the formulas used by PIC and IP to compute the mean velocity $\mathbf{u}$ in a cell for a warm plasma. In PIC, the mean velocity is calculated from:

$$\mathbf{u} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{u_k} = \mathbf{u} + \frac{1}{N}\sum_{k=1}^{N}\mathbf{c_k} \tag{1}$$

where N is the sample size of simulated particles in the cell, $\mathbf{u_k}$ is the velocity vector of particle k that includes a part $\mathbf{c_k}$ due to random thermal motion. In the IP method, the average flow velocity is calculated using:

$$\mathbf{u} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{U_k} \tag{2}$$

where $\mathbf{U_k}$ is the preserved (macroscopic) velocity of particle k.

The PIC method stores $\mathbf{u_k}$, and uses it to compute the particle trajectory and $\mathbf{u}$ according to Eq. (1). The IP method stores both $\mathbf{u_k}$ and $\mathbf{U_k}$. It uses the former to compute the particle trajectory, and uses the latter to compute $\mathbf{u}$ according to Eq. (2). Note that the macroscopic velocity given by the IP formula (2) is the exact value of $\mathbf{u}$ for any sample size, whereas there is the following explicit fluctuation term in the PIC formula (1) that decreases with the square root of the sample size, N:

$$\sigma = \frac{1}{N}\sum_{k=1}^{N}c_k \tag{3}$$

The IP technique has proven very effective in reducing statistical scatter [4-6] in particle simulations of low-speed rarefied gas flows. For example, for benchmark problems such as the Rayleigh flow with a characteristic velocity of 1 m/s, the meaningful velocity and surface shear stress distributions were obtained using the IP technique from a sample size of 10^3–10^4. This sample was four orders of magnitude smaller than that required by DSMC (about 10^8) [4] resulting in a tremendous reduction in CPU time with the IP approach.

An implementation of the IP algorithm for the PIC particle method may be summarized as follows:

1. Assign each simulated particle k an information velocity $\mathbf{U_k}$, density ρ_k, and temperature T_k, and a microscopic (PIC) velocity $\mathbf{u_k}$, and assign each computational cell l a macroscopic velocity $\mathbf{U_l}$, density ρ_l, and temperature T_l.
2. Initially, set $\mathbf{U_l}$, ρ_l, and T_l of each cell according to the initial flow conditions, and set $\mathbf{U_k}$, ρ_k, and T_k of all simulated particles in the cell to be equal to the cell values $\mathbf{U_l}$, ρ_l, and T_l.
3. Move the simulated particles according to the microscopic velocities $\mathbf{u_k}$ using the same algorithms and models as the PIC method that are described in detail in Ref. 1.
4. In a time step Δt, the information velocity of a simulated particle may be changed due to acceleration by external forces. The sum of the forces acting on cell l is

$$F_l = \sum_{j=1}^{2\phi} p_l^j \Delta A_l^j e_l^j + f_l^j \Delta V_l \tag{3}$$

where p_l^j is the gas pressure acting on surface j of cell l with area ΔA_l^j and normal direction e_l^j, ϕ is the flow dimension, and f_l^j is the volume force acting on the cell with volume ΔV_l. The term p_l^j is computed using the pressures in the cells on either side of the surface j. The cell pressures are calculated from the values of ρ_l and T_l using the ideal gas equation of state. The resultant acceleration is:

$$a_l = \frac{F_l}{\rho_l \Delta V_l} \tag{4}$$

which contributes a velocity increment $a_l \Delta t$ to every simulated particle in the cell during the time step.

5. Update $\mathbf{U_l}$, ρ_l and T_l each time step as follows:

   (a) The velocity is updated using

$$\mathbf{U_l} = \omega \mathbf{U_{new}} + (1-\omega)\mathbf{U_{old}} \tag{6}$$

   where ω is a relaxation factor ranging between 0 and 1, $\mathbf{U_{old}}$ is the value of $\mathbf{U_l}$ from the last time step, $\mathbf{U_{new}}$ is the arithmetic mean of the information velocities of all the simulated particles in the cell during this time step:

$$\mathbf{U_{new}} = \frac{\sum_{k=1}^{N} m_k \mathbf{U_k}}{\sum_{k=1}^{N} m_k} \tag{7}$$

   where m_k is the mass of particle k. For large N, ω may be set to 1.

   (b) The density may be updated using the updated cell velocity and the mass conservation equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho U_l)}{\partial x_l} = 0 \tag{8}$$

This equation is differenced using the following implicit scheme:

$$\frac{\rho_l^{i+1} - \rho_l^i}{\Delta t} + \frac{\rho_{l+1}^{i+1} U_{l+1}^{i+1} - \rho_{l-1}^{i+1} U_{l-1}^{i+1}}{2\Delta x} = 0 \tag{11}$$

where superscript i denotes the values at time step i.

   (c) From the kinetic definition of temperature, we have

$$T_{l,i} = \frac{m(\overline{c_i^2} - \overline{c_i}^2)}{k_B} \tag{12}$$

where a bar over a quantity denotes the average value over all the particles in the sample, the subscript i (=1,2,3) denotes the direction, and k_B is the Boltzmann constant.

6. Update the information densities of all the simulated particles in cell l:

$$\rho_k = \rho_l \tag{13}$$

7. Compute macroscopic quantities through statistical averaging of the information quantities.

8. For steady flows, repeat steps 3–6 until the flow reaches a steady state. Then repeat steps (3–7) to sample. For unsteady flows, repeat the steps (2–7) until the end of the evolution period.

The IP steps described above have been incorporated into XES1, an electrostatic, one-dimensional code developed by the Plasma Theory and Simulation Group headed by Professor Birdsall at UC-Berkeley. In the following section, results obtained with the new PIC-IP formulation are compared with those obtained with the original PIC form of the XES1 code.

3. Results

Three different test cases involving warm electrons are considered to allow assessment of the usefulness and accuracy of the PIC-IP scheme: (1) an unperturbed stream of electrons; (2) a perturbed stream of

electrons exhibiting Landau damping; and (3) a two-stream instability flow. These flows are used to compare the PIC and PIC-IP results in terms of their statistical fluctuations, their physical accuracy, and their overall solution time. The codes use normalized input and output data in which time is normalized by the plasma frequency and distance is normalized by the Debye length.

3.1 Unperturbed Stream of Electrons

The problem is initialized with a warm stream of electrons as follows: stream velocity, v_1=3.0; length of system, l=2π; there is no perturbation applied to either the particle velocities or positions, and there is no applied electric field. The numerical parameters are: time step, Δt=0.2; number of computational cells=32; number of iterations=10,000; number of particles per cell is varied.

For this unperturbed system, the steady state solution is a flat profile at the initial velocity, v_1. In Fig. 1a, PIC profiles are shown at the beginning and the end of the simulation that are obtained using 50 particles per cell. These results indicate that the simulation does provide the expected steady state result, but the data are accompanied by a significant degree of statistical fluctuation. These fluctuations are quantified in terms of the standard deviation and listed in Table 1 for a number of different simulations. The effect of increasing the number of particles per cell can be seen in Fig. 1b. The fluctuations are successively reduced as the number of particles is increased such that a one order of magnitude reduction in standard deviation is achieved by increasing the total number of particles by two orders of magnitude, as expected from Eq. (3).

A technique commonly employed in PIC simulations to reduce statistical fluctuations is the so-called "quiet start" in which the initial values of the particle positions are randomized according to the initial particle velocities so as to more uniformly distribute the particles in velocity–position space. The effects of using a quiet start on the unperturbed stream are shown in Fig. 1c. From the figure and from Table 1 we can see that while the quiet start does initially reduce the level of statistical fluctuations at the start of the simulation, the fluctuations levels ultimately reach about the same level as those obtained with the same number of particles without the quiet start.

Finally, in Fig. 1d, the PIC-IP solutions at the beginning and end of the simulation are shown and compared with the corresponding PIC results. As confirmed in Table 1, the level of fluctuations achieved at the end of the PIC-IP simulation using 500 particles per cell is smaller than that obtained with the PIC method using 5,000 particles per cell. For completeness, the PIC-IP and PIC results are also compared for the electric field, and charge density ρ in Figs. 1e and 1f, respectively. These show essentially the same trends as the velocity profiles.

Included in Table 1 are the CPU times for all the simulations. These are the overall run times including all data input and output. The data show that the PIC-IP method using 500 particles per cell obtains solutions with smaller statistical fluctuations than the PIC method with 5,000 particles per cell in a CPU time that is smaller by a factor of more than eight.

Thus, the simple test case of an unperturbed stream shows several important results. PIC simulations

are statistically noisy, and these fluctuations can only be suppressed by employing a significantly larger number of particles. The PIC-IP method successfully reduces these fluctuations and achieves the desired physical solution, allowing significant reductions in computational cost.

3.2 Perturbed Stream of Electrons (Landau Damping)

The flow of the previous section is modified by perturbing the initial particle positions according to $x_1 \cos(2\pi x/l)$ where x_1=0.1, and changing the mean velocity to zero.

Profiles at three different times of charge density, electric field, and stream velocity computed using both the PIC and PIC-IP algorithms are shown in Figs. 2a–2c, respectively. In each simulation, an average of 500 particles per cell is used. In Fig. 2a, the density profile shows a gradual damping from the initial perturbation, and this is also seen in the electric field profile. The stream velocity profile shows an interesting feature where a sinusoidal profile is generated after 10 iterations that eventually is damped back to a nearly flat profile. In all of these plots, it can be seen that there is significant fluctuations in the PIC results, and that the PIC data tend to fluctuate about the smoother PIC-IP profiles. In Fig. 2d, the velocity profile obtained using 5,000 particles per cell with PIC is compared with the PIC-IP solution obtained with 500 particles per cell. In comparison with the PIC data shown in Fig. 2c, the PIC data shown in Fig. 2d exhibit smaller fluctuations and more clearly fluctuate about the PIC-IP solution.

In Fig. 2e, the energy field histories are shown for the PIC and PIC-IP simulations employing 500 particles per cell. The PIC results show the expected behavior of the energy being reduced by more than an order of magnitude before undergoing regular oscillations. As described in Ref. 1, further reduction in the field energy can be achieved with the PIC technique by dividing the velocity distribution function into different species. The values of the field energy predicted by the PIC-IP algorithm are in good general agreement with the PIC results although they appear to level out at a slightly lower energy and oscillate with larger amplitude. It should be noted that these computations are performed without using the quiet start feature.

These comparisons appear to indicate that the PIC-IP algorithm is again able to significantly reduce the statistical fluctuations of the PIC method while maintaining physical accuracy. The overall run time for the PIC-IP simulation using 500 particles per cell is about a factor of seven smaller than that for the PIC simulation employing 5,000 particles per cell. The reduction in computational time obtained using the PIC-IP approach is slightly smaller than that for the simple stream considered in the previous section only because the number of iterations is smaller here and this leads to a proportionally larger time spent in the input and output of data.

3.3 Two-Stream Instability

Consider two opposing streams of electrons with equal number density, n_0, equal temperature, T, and equal velocities moving in opposite directions, $\pm v_1$. There is a background of immobile positive charges with a total density of $2n_0$. When the two streams traverse one wavelength during one plasma oscillation,

a density perturbation in one of the streams is increased by the electrostatic forces created by bunching of particles in the other stream, and vice versa. Hence, the perturbation is observed to grow exponentially with time, and the system is unstable. This type of system is studied here using the following physical parameters: stream velocity, $v_1=\pm 1.0$; thermal velocity (quiet start), $v_t=0.5$; length of system, $l=10\pi$; initial particle positions are perturbed according to $x_1 \cos(2\pi x/l)$ where $x_1=\pm 0.001$. The numerical parameters are: a time step, $\Delta t=0.1$; number of computational cells=128; number of iterations=300; initial number of particles per cell is varied between 64 and 4096.

In Fig. 3a, the charge density profile after completion of one iteration of the PIC solution is shown for various values of average number of particles per cell. Clearly, the initial charge density profile is strongly affected by the number of particles employed with significant reductions in statistical fluctuations obtained when a larger numbers of particles is employed. Similar results are found for the electric field and the mean velocity, as shown in Figs. 3b and 3c, respectively. In the case of the stream velocity, only the profile for the forward moving population of electrons is shown. The population of backward moving electrons has identical behavior. In Fig. 3d, the PIC (4096 particles per cell) and PIC-IP (512 particles per cell) profiles of charge density after one iteration are compared. Clearly, the PIC-IP algorithm is again capable of significantly reducing the fluctuations.

In Figs. 3e through 3g, the density, electric field, and mean velocity profiles obtained at the end of the simulation using various numbers of particles per cell with the PIC method are shown. It is disturbing that the final solutions for all variables depend so strongly on the number of particles employed. These same profiles are compared for the PIC solution employing 4096 particles per cell and the PIC-IP solution employing 512 particles per cell in Figs. 3h through 3j. While there are differences between the solutions, they are also of a more similar nature than the various PIC solutions shown in Figs. 3e–3g. While these types of flow require more careful study, the PIC-IP results are encouraging in that solutions similar to the full PIC method can be obtained with a factor of eight fewer particles (corresponding to an overall reduction in computation time of a factor of six).

4. Conclusions

In this report, we studied the possibility of applying the IP technique to the PIC method. A hybrid PIC/IP algorithm was proposed, and was applied to simulate three different one-dimensional, electro-static problems: (1) an unperturbed electron stream; (b) a perturbed electron stream (landau damping); and (3) a two-stream plasma instability. In each case, the same general conclusions were reached. First, the PIC solutions exhibited strong statistical fluctuations when small numbers of particles were employed. Second, PIC-IP solutions employing factors of 8 to 10 fewer particles exhibited reduced statistical scatter and good general agreement with the best PIC solutions.

This was by no means a complete study and there are several potential directions for further work:

(1) A careful study should be made of the implementation and effect in the PIC-IP algorithm of using additional fluctuation reduction techniques such as the *quiet start* commonly used in PIC.

(2) An electron energy equation should be solved to compute changes in the macroscopic electron temperature.

(3) Cold plasma conditions should be studied.

(4) The PIC-IP algorithm should be extended to include magnetic fields, and multi-dimensional flows.

5. References

[1] Birdsall, C.K., and Langdon, A.B., *Plasma Physics via Computer Simulation,* Hilger, New York, 1991.

[2] Bird, G.A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows,* Clarendon, Oxford, 1994.

[3] Oran, E.S., Oh, C.K., and Cybyk, B.Z., "Direct simulation Monte Carlo: Recent advances and application," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, p. 403.

[4] Fan, J., and Shen, C., "Statistical simulation of low-speed unidirectional flows in transition regime," in *Rarefied Gas Dynamics*, edited by R. Brun, Cepadues, Vol. 2, 1999, p. 245.

[5] Cai, C.P., Boyd, I.D., Fan, J., and Candler, G.V., "Direct simulation methods for low-speed microchannel flows" *Journal of Thermophysics and Heat Transfer,* Vol. 14, 2000, p. 368.

[6] Fan, J., Boyd, I.D., Cai, C.P., Hennighausen, K., and Candler, G.V., "Computation of rarefied gas flows around a NACA 0012 airfoil," *AIAA Journal,* Vol. 39, 2001, p. 618.

Table 1. Numerical properties for simulations of an unperturbed electron beam.

Method	Particles Per Cell	Iteration	σ	CPU (sec)
PIC	50	10,000	0.065	25
PIC	500	10,000	0.016	103
PIC	5,000	10,000	0.0066	915
PIC (quiet start)	50	1	0.022	0.3
PIC (quiet start)	50	10,000	0.079	25
PIC-IP	500	10,000	0.0054	115

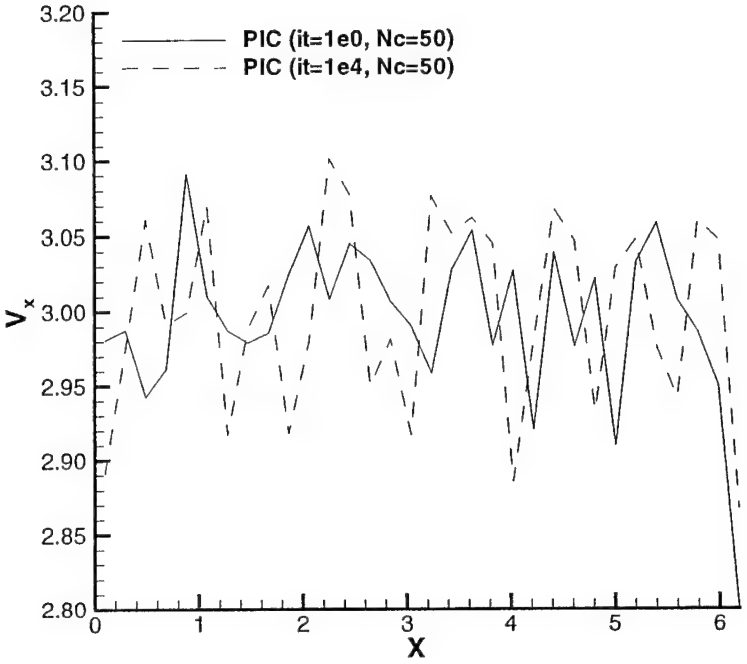


Fig. 1a. Velocity profile for unperturbed stream.

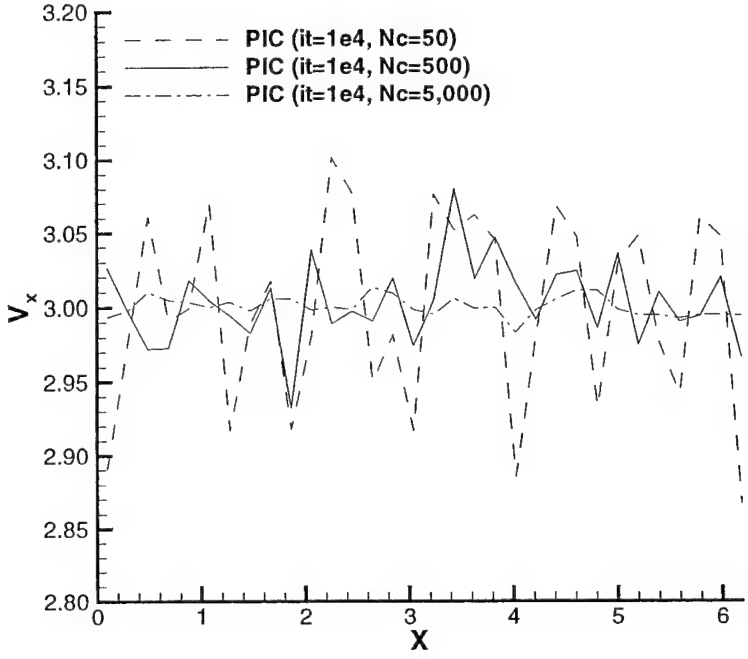


Fig. 1b. Velocity profile for unperturbed stream.

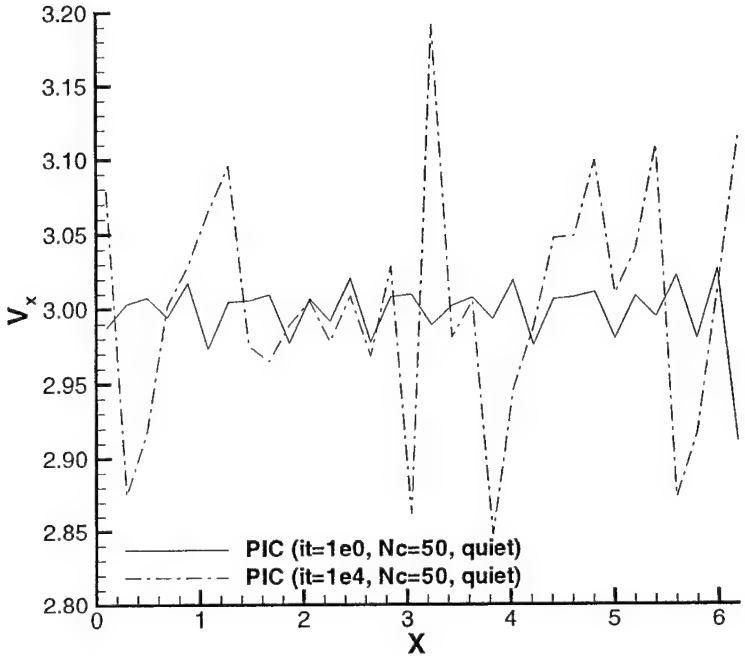


Fig. 1c. Velocity profile for unperturbed stream.

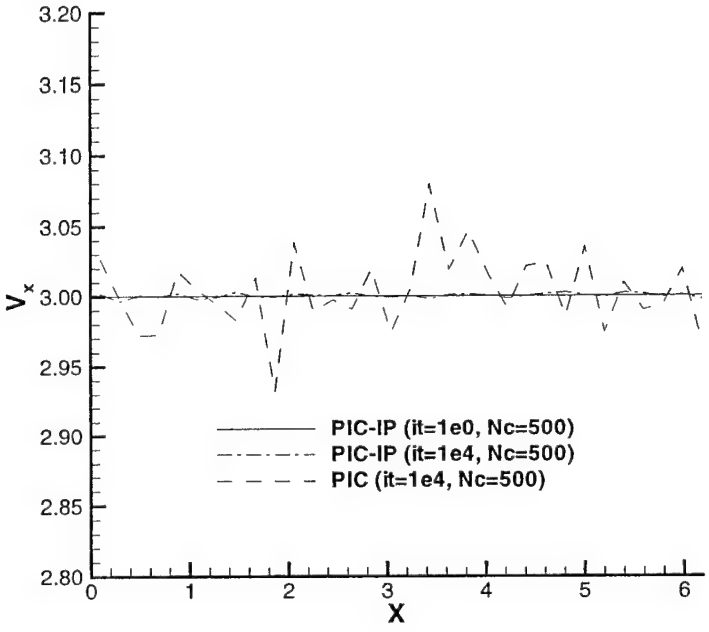


Fig. 1d. Velocity profile for unperturbed stream.

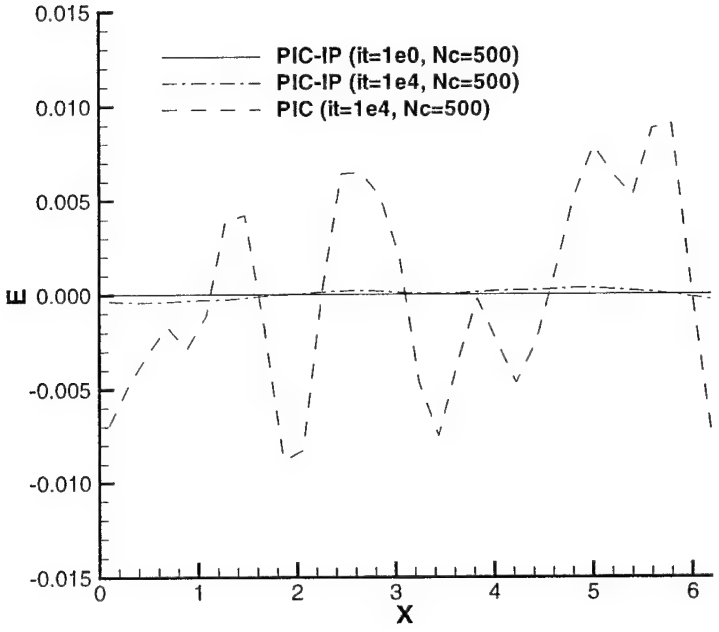


Fig. 1e. Electric field profile for unperturbed stream.

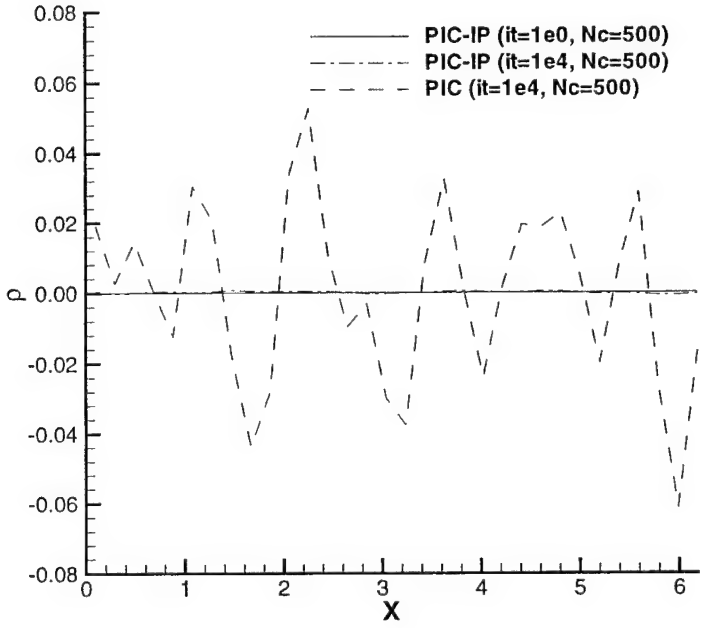


Fig. 1f. Density profile for unperturbed stream.

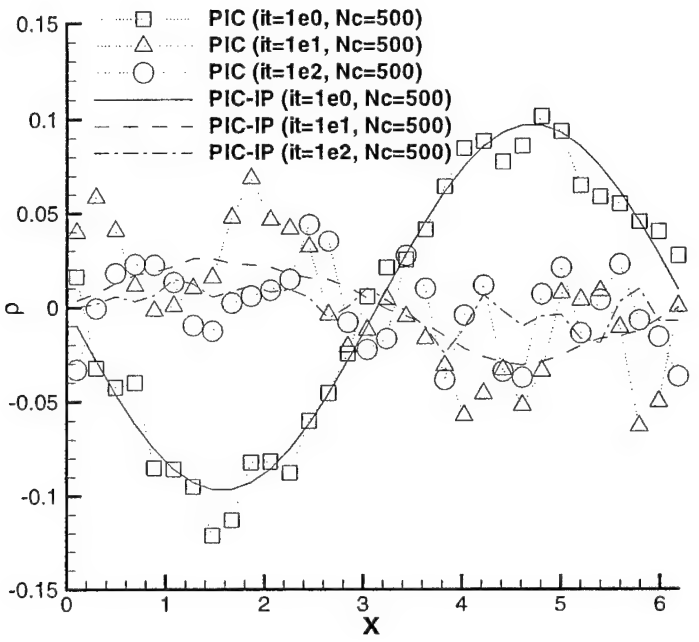


Fig. 2a. Density profile for perturbed stream.

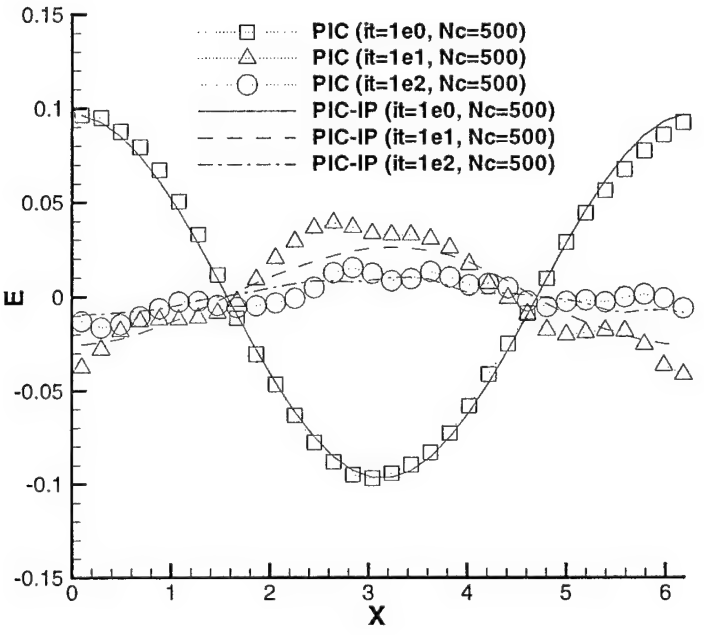


Fig. 2b. Electric field profile for perturbed stream.

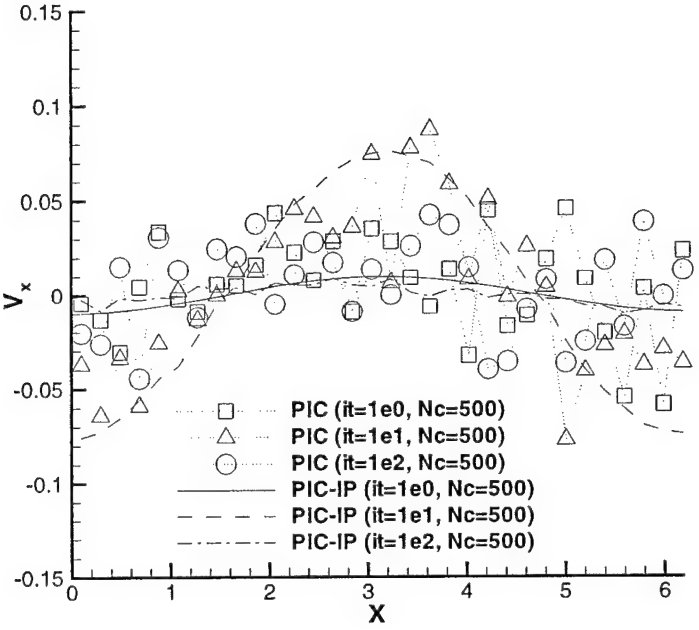


Fig. 2c. Velocity profile for perturbed stream.

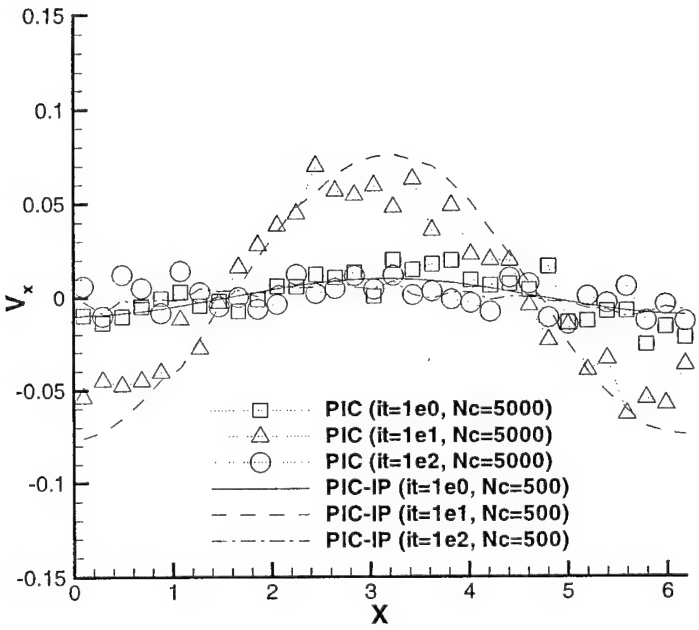


Fig. 2d. Velocity profile for perturbed stream.

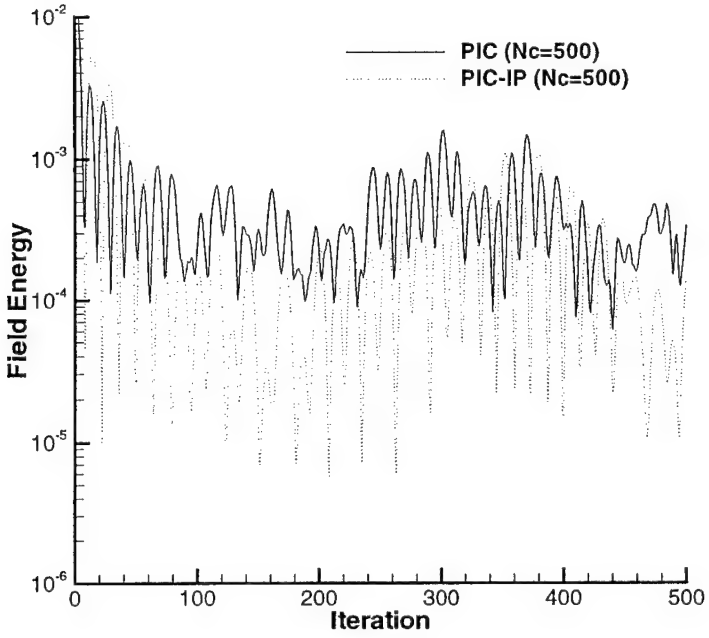


Fig. 2e. Field energy history for perturbed stream.

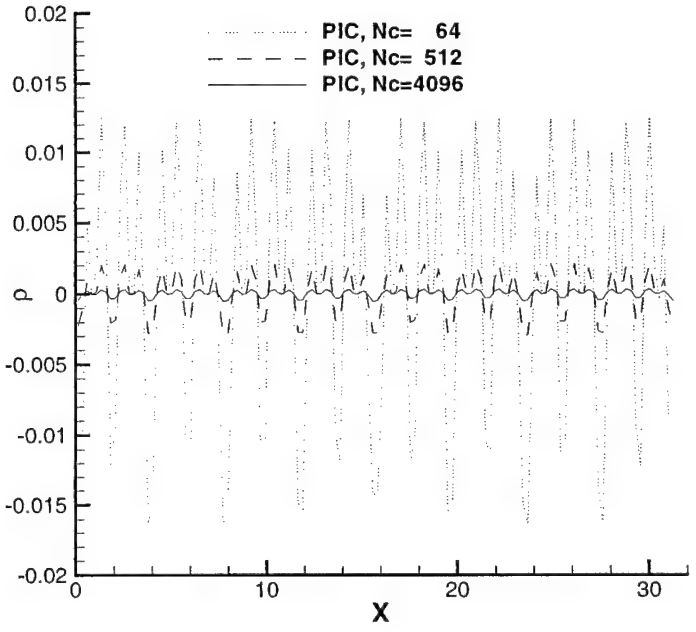


Fig. 3a. Initial density profiles for warm, two-stream instability.

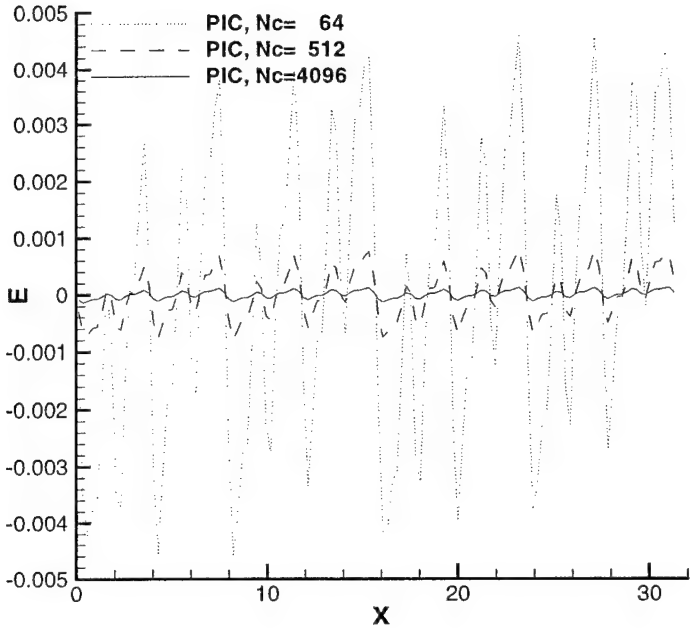


Fig. 3b. Initial electric field profiles for warm, two-stream instability.

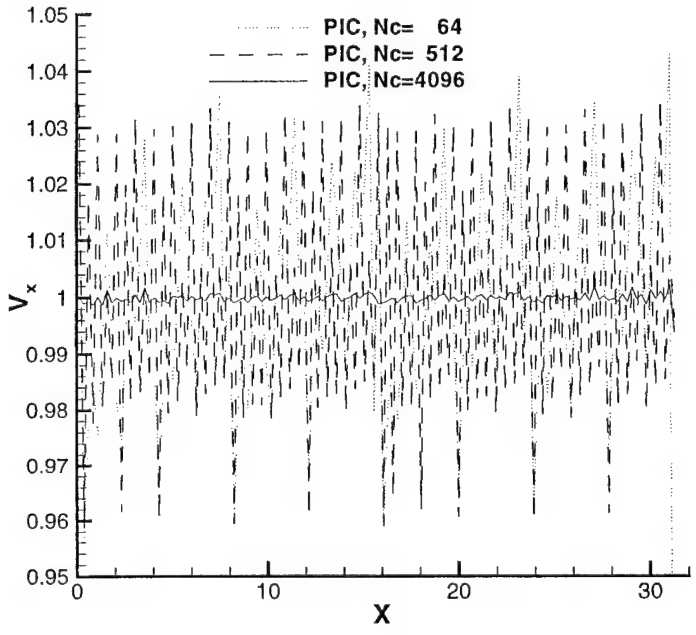


Fig. 3c. Initial velocity profiles of forward moving electron population for warm, two-stream instability.

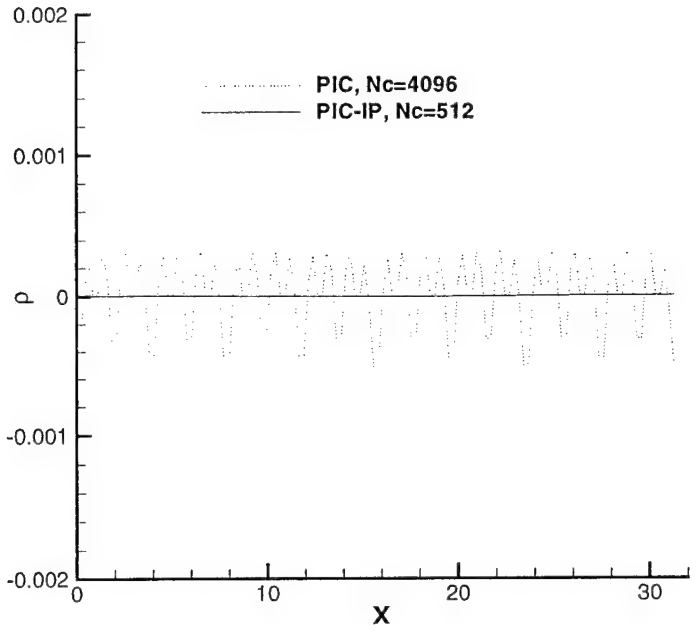


Fig. 3d. Initial density profiles for warm, two-stream instability.

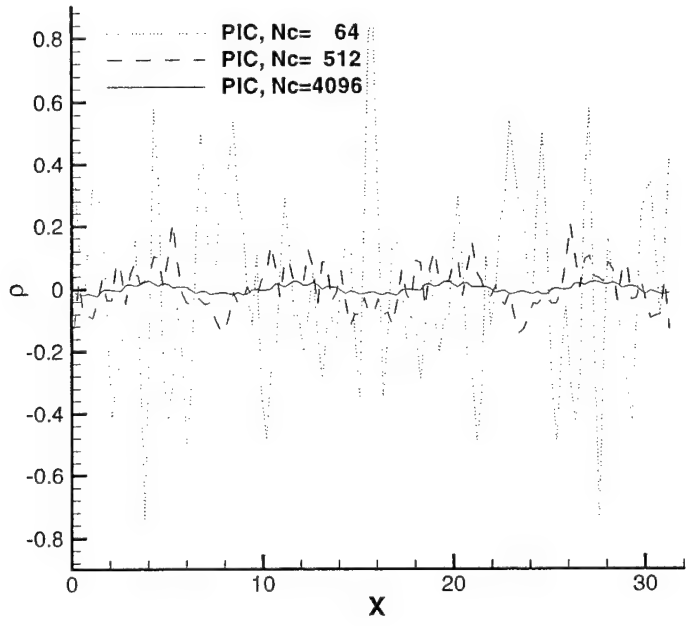


Fig. 3e. Final density profiles for warm, two-stream instability.

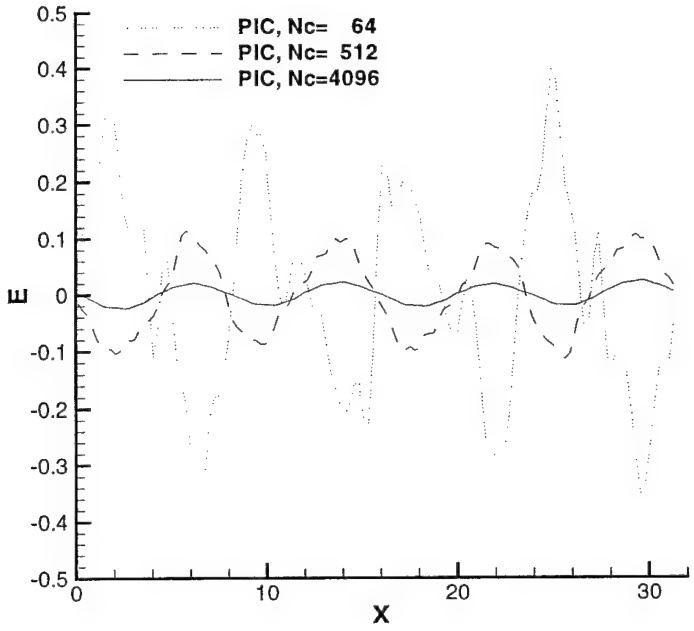


Fig. 3f. Final electric field profiles for warm, two-stream instability.

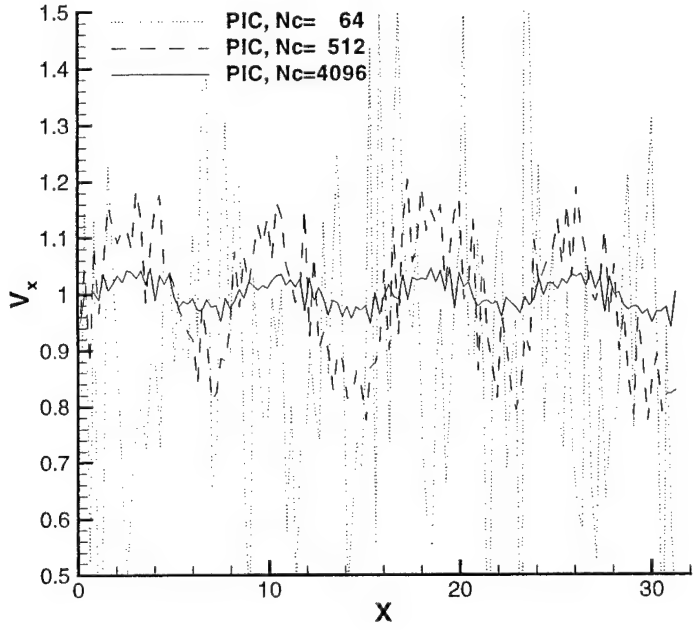


Fig. 3g. Final velocity profiles of forward moving electron population for warm, two-stream instability.

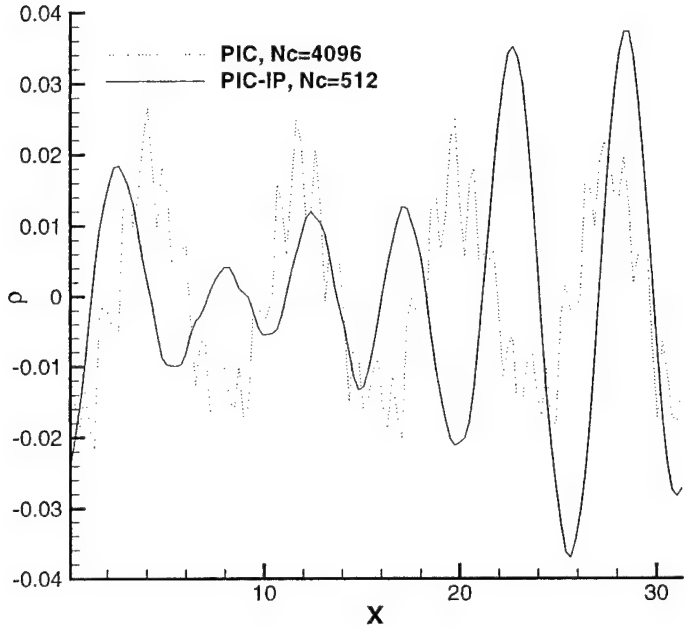


Fig. 3h. Final density profiles for warm, two-stream instability.

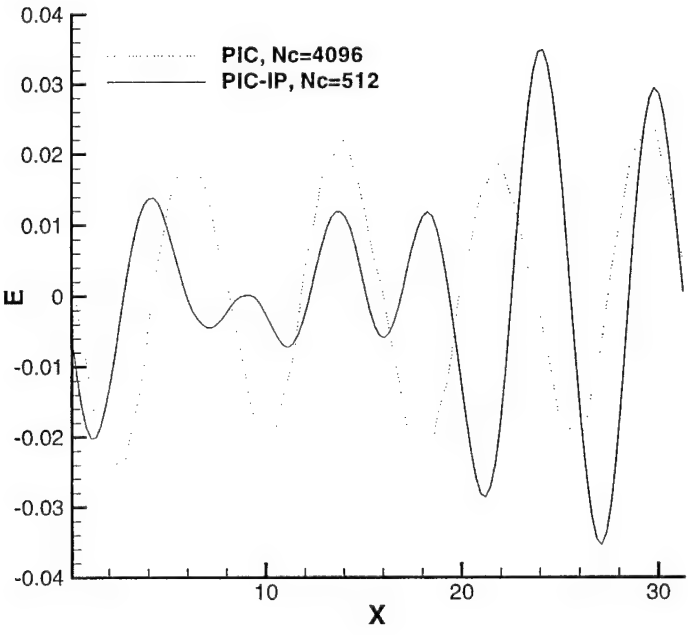


Fig. 3i. Final electric field profiles for warm, two-stream instability.

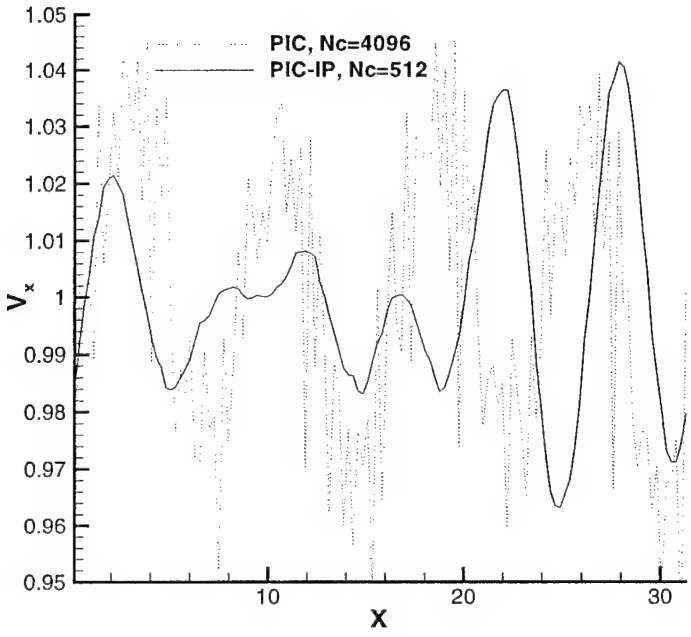


Fig. 3j. Final velocity profiles of forward moving electron population for warm, two-stream instability.

A radiation damping algorithm for particle simulation

X. Yu, J. P. Verboncoeur, and D. Bruhwiler*
Dept. Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720-1770
johnv@eecs.berkeley.edu

DRAFT: October 11, 2000

Abstract

Radiation damping is a phenomenon that occurs when a charged particle undergoing a large acceleration emits radiation. While this occurs in a linear system only with extreme accelerations, the effect is more pronounced for a magnetized acceleration where the radial acceleration induced by the Lorentz force is sufficient to generate synchrotron radiation. The effect is only measurable in highly relativistic charged particles. The radiation damping results in loss of energy and momentum, so that an electron orbit will decay in time. In this work, we consider a relativistic electron orbiting in a uniform magnetic field. A model following Jackson [1] is implemented in the oarticle-in-cell code XOOPIC [2]. The model includes the momentum and energy loss effects, but neglects the propagation of the emitted radiation. Simulation results are obtained for a 10 GeV electron in a 75 T field, and the temporal decay of the energy compares favorably to theory.

*TechX Corporation, 1280 28th St., Suite 2, Boulder, CO 80303

1 Introduction

Radiation damping is a physical phenomenon in which a charged particle undergoing rapid acceleration emits radiation, resulting in loss of particle energy and momentum. The rapid acceleration may be related to forces from electromagnetic fields or from collisions. This effect, sometimes referred to as self-force, is only of significance for highly relativistic charged particles undergoing extreme acceleration.

While collisions can also result in significant accelerations, the energy loss associated with a collision can be treated as part of the collision mechanics; we are seeking effects here that modify the traditional Newton-Lorentz equations of motion. In addition to collisions, there are two general classical cases where radiation damping is exhibited. One is linear acceleration of electrons in a strong electric field. The other is synchrotron radiation emitted by electrons accelerating in due to circular motion about a magnetic field. For the linear acceleration case, the energy loss is small for most achievable parameters, and is thus not directly addressed here. Synchrotron radiation, on the other hand, is readily observed and is a significant factor in the design of particle accelerators. In this research, the case of a particle accelerating in circular motion about a uniform static magnetic field is considered, since this creates a classical acceleration of magnitude sufficient to induce significant radiation damping.

A macroscopic view of radiation damping is discussed by Jackson [1], including the linear acceleration and synchrotron cases as well as the angle and spectrum of the emitted radiation. Yaghjian [3] presents a detailed study for a spherically shaped particle, and Rossi [2] presents a brief discussion on the topic. The general conclusions of the authors are similar, although the complexity and level of detail varies greatly. This study focuses on the Jackson model due to its simplicity and its applicability to the PIC model. The authors are not aware of any other computational work in the literature attempting to address radiation damping.

The objective of the research was to implement and analyze a computationally efficient correction to the standard particle-in-cell (PIC) scheme to account for the particle kinetics when radiation damping becomes non-negligible. This work makes the PIC algorithms more accurate for modeling high-energy particles undergoing rapid accelerations by modifying the equations of motion to account for the loss of energy. The radiated energy is not

treated here; in particular, the spectral and directional distribution of the radiated energy is not considered, and the interaction of the radiated fields with the particle is also neglected; the radiated energy is simply treated as lost energy.

The theoretical model employed is discussed in Section 2, including a discussion of the computational resources required to include the radiated energy and fully resolve the generated fields and their effects on the particle.
***

2 Radiation Damping Model

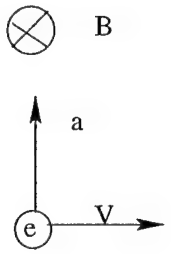


Figure 1: The Model used for the algorithm.

In this section, a we start with a general model for radiation damping from Jackson [1]. The linear acceleration is discussed, and it is demonstrated that the effect is negligible for this case. Next, an electron accelerating in circular motion in a static homogenous magnetic field is considered, again following Jackson. The power loss is described for the synchrotron case, and the spectral and direction emission of radiation is discussed, as well as the ramifications of modeling the emitted radiation in a PIC code. Finally, the radiation damping algorithm is discretized for incorporation into the PIC model.

For an electron undergoing a rapid acceleration, Jackson related the radiated power to the derivative of the electron momentum:

$$P = \frac{2}{3}\frac{e^2}{4\pi\varepsilon_0 m^2 c^3}\left[\left(\frac{d\mathbf{p}}{d\tau}\right)^2 - \beta^2\left(\frac{dp}{d\tau}\right)^2\right], \tag{1}$$

where e is the electronic charge, m is the electron rest mass, c is the speed of light, ε_0 is the permittivity of free space, $d\tau = dt/\gamma$ is the proper time element, $\gamma = (1-\beta^2)^{-1/2}$, $p = \gamma m v$, $\beta = v/c$, and v is the electron velocity.

For linear acceleration, the corresponding result is written:

$$P = \frac{2}{3}\frac{e^2}{4\pi\varepsilon_0 m^2 c^3}\left(\frac{dp}{dt}\right)^2. \tag{2}$$

It is instructive to note that in the linear acceleration case, Jackson showed that the radiation losses are negligible compared to the effects of the external forces acting on the electron when the change in energy per unit distance satisfies

$$\left|\frac{dE}{dx}\right| < \left(\frac{2}{3}\frac{e^2}{4\pi\varepsilon_0 m^2 c^4 \beta}\right)^{-1}, \tag{3}$$

which becomes 2.7×10^{20} V/m for highly relativistic electrons. Since this is far above the gradients of 10^7 V/m present in linear accerators, radiation damping is a negligible effect for linear devices. We therefore consider only the synchrotron model in this research.

Consider an electron in circular motion in a static magnetic field as shown in Fig. 1. For synchrotron motion, the radiated power becomes [1]:

$$P = \frac{2}{3}\frac{e^2}{4\pi\varepsilon_0 r_L^2}\beta^4\gamma^4, \tag{4}$$

and the relativistic Larmor radius is given by:

$$r_L = \frac{v}{\omega_c} = \frac{\gamma m \beta c}{eB} = \sqrt{\gamma^2 - 1}\frac{mc}{eB}, \tag{5}$$

where B is the external static magnetic field.

The spectral content of the radiation power is a complicated function. The radiation is strongly polarized in the plane of motion, with increasing confinement to the plane of motion with increasing frequency. The radiation power is confined to increasingly smaller angles at higher frequencies, as well as with increasing particle energy. Jackson derived the energy radiated per unit frequency per unit solid angle for the synchrotron case:

$$\frac{d^2 I}{d\omega d\Omega} = \frac{e^2}{3\pi^2 c}\left(\gamma^{-2}+\theta^2\right)^2\left(K_{2/3}^2(\xi) + \frac{\theta^2}{\gamma^{-2}+\theta^2}K_{1/3}^2(\xi)\right), \tag{6}$$

where θ is the angle from the plane of circular motion, K is the modified Bessel function, and ξ is given by:

$$\xi = \frac{\omega r_L}{3c}\left(\gamma^{-2} + \theta^2\right)^{3/2}. \tag{7}$$

The radiation is negligible beyond the critical frequency,

$$\omega_{crit} = 3\gamma^3 \frac{c}{r_L}. \tag{8}$$

The number of harmonics present in the radiation spectrum for periodic motion of angular frequency ω is given by $n_{crit} = 3\gamma^3$. The critical angle beyond which the radiated power drops to $1/e$ of the peak value is given by

$$\theta_{crit} \approx \frac{1}{\gamma}\sqrt{\frac{\omega_{crit}}{3\omega}}. \tag{9}$$

From Eq. 9, it is evident that the radiated power is confined to increasingly smaller angles at higher frequencies, as well as increasing particle energy. The radiation is emitted from the particle undergoing circular motion in a pattern similar to the sweep of a lighthouse beam, with the focus in the direction of the velocity vector.

To assess the time and space scales required to resolve the emitted radiation, consider a 1 GeV electron orbiting a 10 T static magnetic field. The relativistic momentum factor is $\gamma = 1952$, and the velocity is effectively $v = c$. The gyroradius is $r_L = 0.334$ m. The power radiated is $P = 6 \times 10^{-6}$ W, so the electron loses about 1% of its energy in 270 ns. The critical frequency $\omega_{crit}/2\pi = 3.2 \times 10^{18}$ Hz, with a corresponding critical wavelength of $\lambda_{crit} = 9.4 \times 10^{-11}$ m. In order to resolve the wavelength, the mesh size requirement becomes $\Delta x \leq \lambda/4 = 2.4 \times 10^{-11}$ m. For a nominal system length of $L = 3r_L$ in each dimension, the required number of cells in each dimension is 4.2×10^{10}, or 1.7×10^{21} cells in a two-dimensional simulation. The storage required for this spatial resolution exceeds the memory capacity of exisiting computers. Furthermore, the timestep required to satisfy the Courant condition is $\Delta t \leq \Delta x/\sqrt{2}c = 5.7 \times 10^{-20}$ s. The cyclotron period is 7 ns, so one orbit would require 1.2×10^{11} timesteps, and running to the 1% energy loss would require 4.7×10^{12} timesteps. Resolving the critical frequency requires a timestep $\Delta t \leq 3.1 \times 10^{-19}$ s, so frequency resolution is even more untenable. It is clear from these space and time constraints that it will remain

impossible for the forseeable future to simulate the self-consistent emission of radiation from even a single particle within a standard PIC scheme. It is equally impossible to simulate the emitted fields over a frequency spectrum sufficient to include a significant fraction of the radiated power.

The total power radiated for the synchrotron case can be written by modification of Eq. 1:

$$P = \frac{e^2\gamma^2}{6\pi\varepsilon_0 c^3}\left[\left(\frac{d\mathbf{u}}{dt}\right)^2 - \beta^2\left(\frac{du}{dt}\right)^2\right] \tag{10}$$

where $\mathbf{u} = \gamma\mathbf{v}$ and $u = \sqrt{\mathbf{u}\cdot\mathbf{u}}$.

Using $-P = (dE/dt)_{rad}$ for the radiation energy loss rate, where the particle energy is given by $E = \gamma mc^2$, we can rewrite the power loss:

$$\left(\frac{d\gamma}{dt}\right)_{rad} = -\frac{1}{mc^2}P = -\frac{e^2\gamma^2}{6\pi\varepsilon_0 mc^5}\left[\left(\frac{d\mathbf{u}}{dt}\right)^2 - \beta^2\left(\frac{du}{dt}\right)^2\right] \tag{11}$$

Writing the magnitude of the normalized momentum in terms of γ:

$$u = c\sqrt{\gamma^2 - 1}, \tag{12}$$

and taking the time derivative to obtain

$$\frac{du}{dt} = \frac{\gamma c}{\sqrt{\gamma^2 - 1}}\frac{d\gamma}{dt}. \tag{13}$$

Combining Eqs. 11 and 13, and applying the result to radiation damping:

$$\left(\frac{du}{dt}\right)_{rad} = -\frac{1}{6\pi\varepsilon_0}\frac{e^2}{mc^4}\frac{\gamma^3}{\sqrt{\gamma^2 - 1}}\left[\left(\frac{d\mathbf{u}}{dt}\right)^2 - \left(\frac{u}{\gamma c}\right)^2\left(\frac{du}{dt}\right)^2\right]. \tag{14}$$

Discretization goes here [David Bruhwiler]

3 Analysis and Verification of the Algorithm

In order to verify that the damping rate obtained in the simulation is correct, an analytical formula is derived for the energy as a function of time, and is compared to the simulation result for the synchrotron radiation case.

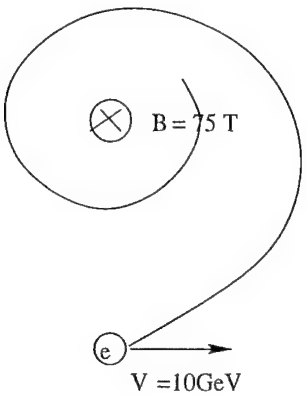


Figure 2: Synchrotron decay of single electron orbit.

We can write variables in terms of energy, $\gamma = E/mc^2$, as well as $\beta^2 = 1 - m^2c^4/E^2$. The particle velocity can be written

$$v = c\sqrt{1 - \frac{m^2c^4}{E^2}}, \tag{15}$$

so the normalized momentum becomes

$$u = c\sqrt{\frac{E^2}{m^2c^4} - 1}. \tag{16}$$

For highly relativistic particles, Eq. 16 reduces to $u \approx E/mc$.

Then we can write

$$\frac{du}{dt} = \frac{du}{dE}\frac{dE}{dt} = \frac{1}{mc\sqrt{1 - m^2c^4/E^2}}\frac{dE}{dt} \tag{17}$$

When the energy is large compared to the rest energy,

$$\frac{du}{dt} \approx \frac{1}{mc}\frac{dE}{dt} \tag{18}$$

Consider the equation of motion for an electron in a magnetic field:

$$\frac{d\mathbf{p}}{dt} = \frac{md\mathbf{u}}{dt} = -e\mathbf{v} \times \mathbf{B}. \tag{19}$$

Here we consider the synchrotron case shown in 2, where the magnetic field is perpendicular to the plane of motion. Using Eq. 15,

$$\frac{du}{dt} = \frac{eB}{m} c \sqrt{1 - \frac{m^2 c^4}{E^2}}. \tag{20}$$

In the high-energy limit,

$$\frac{du}{dt} = \frac{eBc}{m}. \tag{21}$$

Using Eq. 21 in Eq. 10, we obtain:

$$\frac{dE}{dt} = -\frac{1}{6\pi\varepsilon_0} \frac{e^2}{c^3} \frac{E^2}{m^2 c^4} \left[\frac{e^2 B^2}{\varepsilon_0 \mu_0 m^2} - \frac{1}{m^2 c^2} \left(\frac{dE}{dt} \right)^2 \right] \tag{22}$$

To approximate the equation, compare $\frac{e^2 B^2}{\epsilon_0 \mu_0}$ with $\frac{1}{c^2}(\frac{dE}{dt})^2$. Insert constants and parameters for electrons, we get

$$\frac{e^2 B^2}{\epsilon_0 \mu_0} = \frac{(1.6 \times 10^{-19})^2 \times 75^2}{8.85 \times 10^{-12} \times 4\pi \times 10^{-7}} = 1.29 \times 10^{-17}$$

$$\frac{1}{c^2}(\frac{dE}{dt})^2 \approx \frac{1}{(3 \times 10^8)^2}(10^{-1})^2 = 1.1 \times 10^{-19}$$

where we used approximation $\frac{dE}{dt} \approx 0.1$, which is valid as seen from the next sections. And this approximation will keep being valid since the energy radiation rate will decrease. So the second term in equation(12) could be dropped for approximation.

Thus, the approximated form becomes:

$$\frac{dE}{dt} = -\frac{1}{6\pi\epsilon_0} \frac{e^2}{c^3} \frac{E^2}{m^2 c^4} \frac{e^2 B^2}{\epsilon_0 \mu_0 m^2} = -\frac{e^4 B^2 E^2}{6\pi \epsilon_0^2 \mu_0 m^4 c^7}$$

$$\frac{dE}{dt} = -\frac{(1.6022 \times 10^{-19})^4 \times 75^2 \times E^2}{6\pi (8.85 \times 10^{-12})^2 4\pi \times 10^{-7} \times (9.1095 \times 10^{-31})^4 \times (2.9979 \times 10^8)^7}$$

$$\frac{dE}{dt} = -1.3191 \times 10^{16} E^2 \tag{23}$$

Solving for,

$$E = \frac{1}{\frac{1}{E_0} + 1.3191 \times 10^{16} t} \tag{24}$$

or,

$$\frac{1}{E} = \frac{1}{E_0} + 1.3191 \times 10^{16} t \tag{25}$$

4 Results

As stated before, synchrotron radiation could give a good observation of the damping. An input file was set up to simulate it. In this simulation, one electron is injected into a strong magnetic field with a large initial energy.

Attachment 1 gives the input file used. And figure 3 shows the x-y phase space of the movement of the electron. And the energy damping could be easily observed from the decreasing of its gyro-radius.

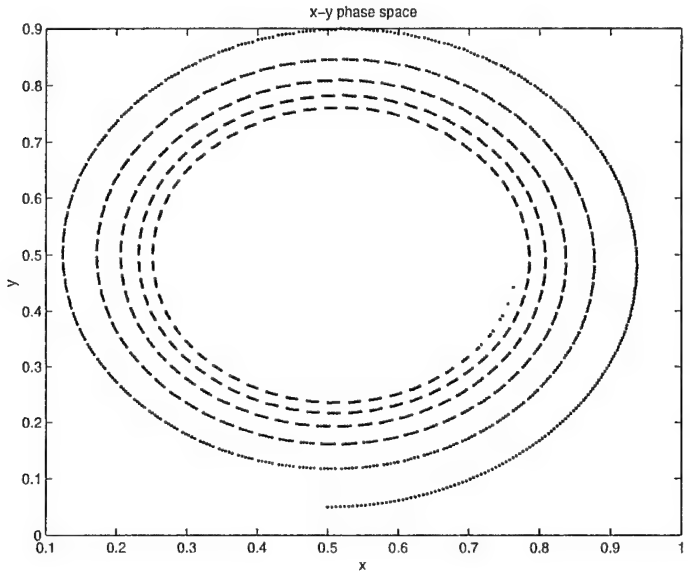


Figure 3: Electron position trace from simulation.

And the verification of the simulation was done by comparing the kinetic energy diagnostic with the energy damping function developed above. And a comparason of the analytical energy function with the energy loss function from Jackson's was first done to illustrate its correctness.

4.1 Check the energy formula against Jackson's one circle energy dissipation equation.

My formula (13) is in MKSA. Check it against Jackson's equation (14.33) for high energy electrons ($\beta \approx 1$):

$$\delta E(Mev) = 8.85 \times 10^{-2} \frac{[E(Gev)]^4}{\rho(\text{meters})}$$

For the first circulation, gyration radius(in relativistic form) is given by:

$$\rho = \frac{\gamma m v_0}{eB} = \frac{1.9570 \times 10^4 \times 9.1095 \times 10^{-31} \times 2.9979 \times 10^8}{1.6022 \times 10^{-19} \times 75} = 0.4448 \quad (m)$$

Where v_0 is approximated to be light velocity, since $v_{1drift} = 10^{10} ev$ as given in the input file. And γ is calculated as following:

Kinetic energy:

$$T = (\gamma - 1)mc^2 = eV$$

$$\gamma = 1 + \frac{eV}{mc^2} = 1 + \frac{1.6022 \times 10^{-19} \times 10^{10}}{9.1095 \times 10^{-31} \times (2.9979 \times 10^8)^2} = 1.9570 \times 10^4$$

Then the time for the particle to move through the first cycle is calculated by

$$t = \frac{2\pi\rho}{v} \approx \frac{2\pi\rho}{c} = \frac{2 \times 3.14 \times 0.4448}{2.9979 \times 10^8} = 9.3224 \times 10^{-9} \quad (s)$$

Calculate δE by my formula:

$$\frac{1}{E} = \frac{1}{E_0} + 1.3191 \times 10^{16} t = \frac{1}{1.6 \times 10^{-9}} + 1.3191 \times 10^{16} \times 9.3224 \times 10^{-9} = 747113583$$

$$\delta E = E - E_0 = \frac{1}{747113583} - 1.6022 \times 10^{-9} = -2.6152 \times 10^{-10} \quad (Joules)$$

Calculate δE by Jackson's:

$$\begin{aligned}
\delta E_{max} &= -8.85 \times 10^{-2} \frac{E_0^4}{\rho} \\
&= -8.85 \times 10^{-2} \times \frac{(10)^4}{0.4448} \\
&= -1989.6 Mev \\
&= -3.1874 \times 10^{-10} \quad (jourls)
\end{aligned}$$

$$\begin{aligned}
\delta E_{min} &= -8.85 \times 10^{-2} \frac{E^4}{\rho} \\
&= -8.85 \times 10^{-2} \times \frac{(10 - 1.9896)^4}{0.4448} \\
&= -819.21 Mev \\
&= -1.3125 \times 10^{-10} \quad (joules)
\end{aligned}$$

The result from analytical formula falls into the interval between the maximum and minimum energy change. Noticing the Jackson's euqation is also an approximation, this suggests the correctness of my formula.

4.2 Check simulation against analytical energy-time function.

Pick two points from KE diagonistic.

$$t_1 = 1.345 \times 10^{-8}, E_1 = 0.0001247 \times 10^{-5}$$

Using $E(t)$ function, $E = 1.2476 \times 10^{-9}$. And the difference is -0.048%.

$$t_2 = 1.4948 \times 10^{-6}, E_2 = 4.9986 \times 10^{-11}$$

Using $E(t)$ function, $E = 4.9214 \times 10^{-11}$. And the difference is 0.091%.

We are using $\frac{E_s - E_t}{E_s}$ as defference. And all the values above are in MKSA.

As we can see,the simulation result and the analytical result are very close to each other. Attatchment 2, 3, 4, 5 show the curves from simulation, from analytical results, a combination of them and the difference between them.

5 Conclusion

As shown by the analysis above, the radiation damping algorithm implemented to PIC code is an accurate algorithm to approach the problem. This makes PIC a more powerful tool in various research work related to plasma simulation.

References

[1] J. D. Jackson. *Classical Electrodynamics.* John Wiley and Sons, 1975.

[2] B. Rossi. *High Energy Particles.* Prentice-Hall, 1952.

[3] A. D. Yaghjian. *Relativistic Dynamics of a Charged Sphere.* Springer-Verlag, 1992.

Figure 4: XOOPIC input file for simulation.

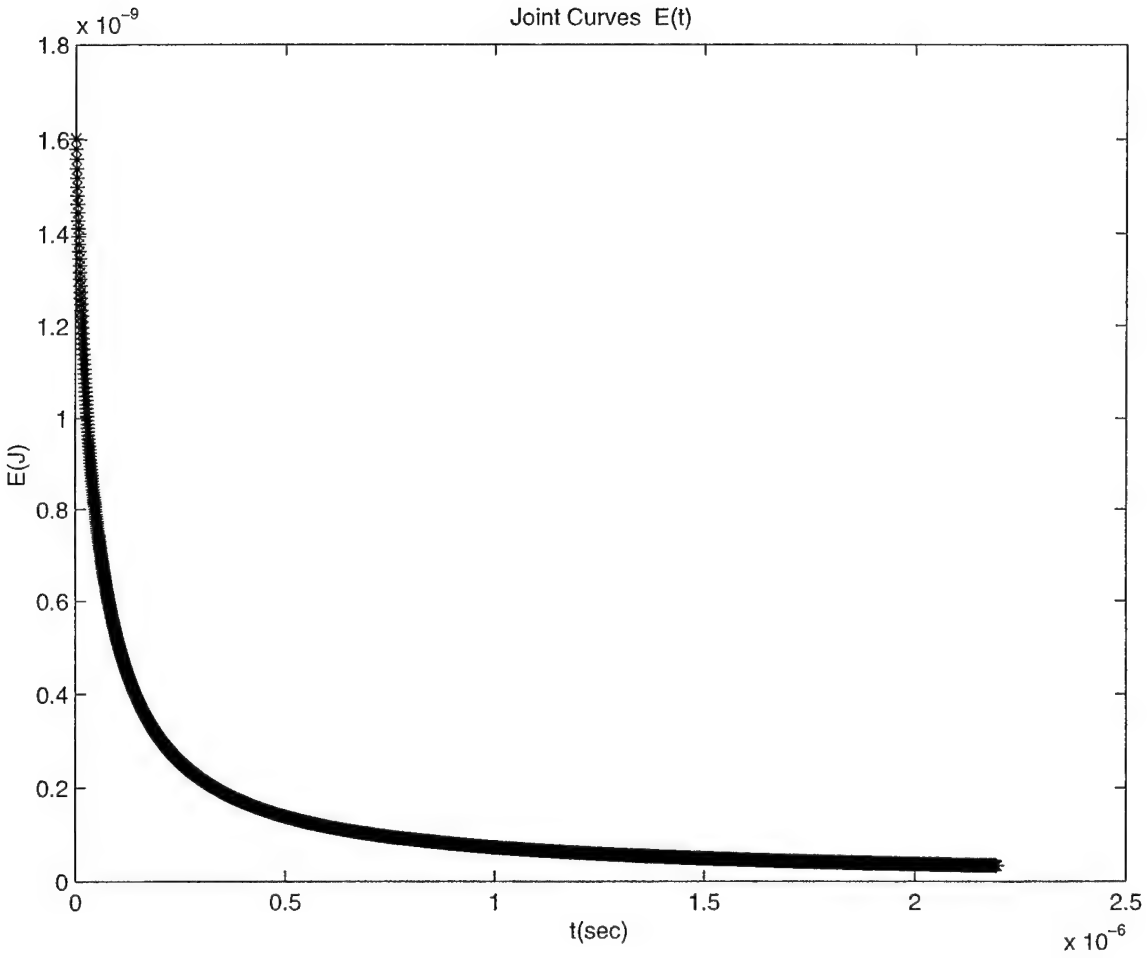


Figure 5: Electron energy decay from analytic result *solidline* and simulation *symbols*.

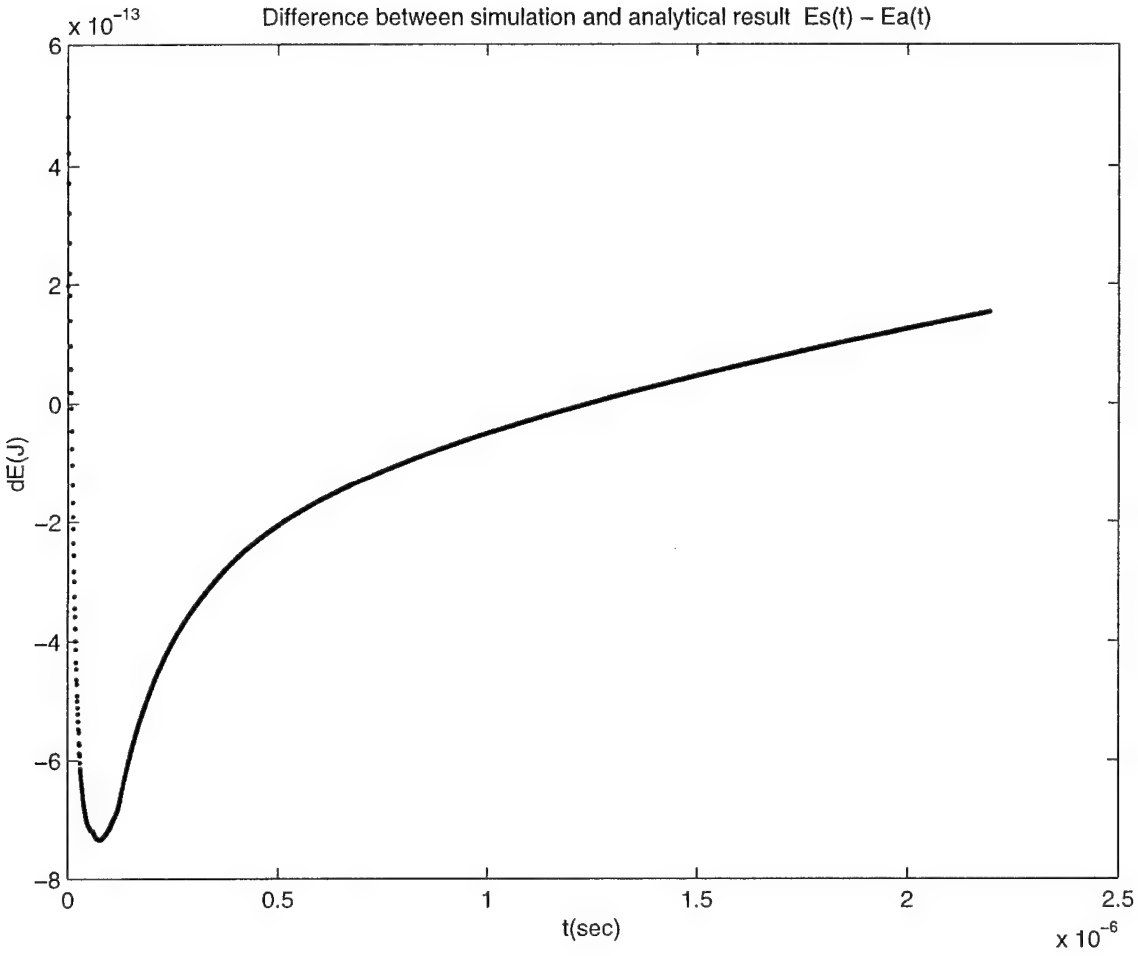


Figure 6: Difference in energy decay between theory and simulation.

from IEEE ICOPS Simulation Minicourse (Monterey, CA, June 1999)

J. P. Verboncoeur
NE and EECS Depts.
University of California
Berkeley, CA 94720-1770

1 Secondary Emission

The process of electron impact secondary emission is a key element of a number of processes related to high power microwave tubes. Secondary emission plays a pivotal role in depressed collectors, single- and two-surface multipactors, and beam interception. In this section, a secondary emission model is outlined, and the implementation in the XOOPIC PIC-MCC code [1] is described.

Electron impact secondary emission occurs when an electron impacts a surface, which may be a conductor or a dielectric, and ejects electrons from the surface. In PIC codes, it is not possible to model the quantum mechanical details of the process, which involves interaction of the incident electron with conduction or valence band electrons in the surface medium, due to the time and space scales involved. Instead, it is more efficient to employ a phenomenological model.

1.1 Energy and Angular Dependence

This work is based on the secondary model due to Vaughan [2] and later experimentally verified by Shih [4], and improved by Vaughan [3] and later [5]. The secondary electron coefficient, defined as the ratio of ejected to incident electrons, has both energy and angular dependence:

$$\delta(\mathcal{E},\theta) = \delta_{\max 0}\left(1 + k_{s\delta}\frac{\theta^2}{2\pi}\right)(w\exp(1-w))^k . \tag{1}$$

Here, the incident energy is given by $\mathcal{E}$, the angle with respect to the surface normal is θ, $k_{s\delta}$ is a surface smoothness parameter described below, k is a curve-fit parameter also described below, and $\delta_{\max 0}$ is the peak coefficient, which occurs at normal incidence at the energy $\mathcal{E}_{\max 0}$. The energy dependence appears implicitly in the right hand side of Eq. 1 through the normalized energy, w, given

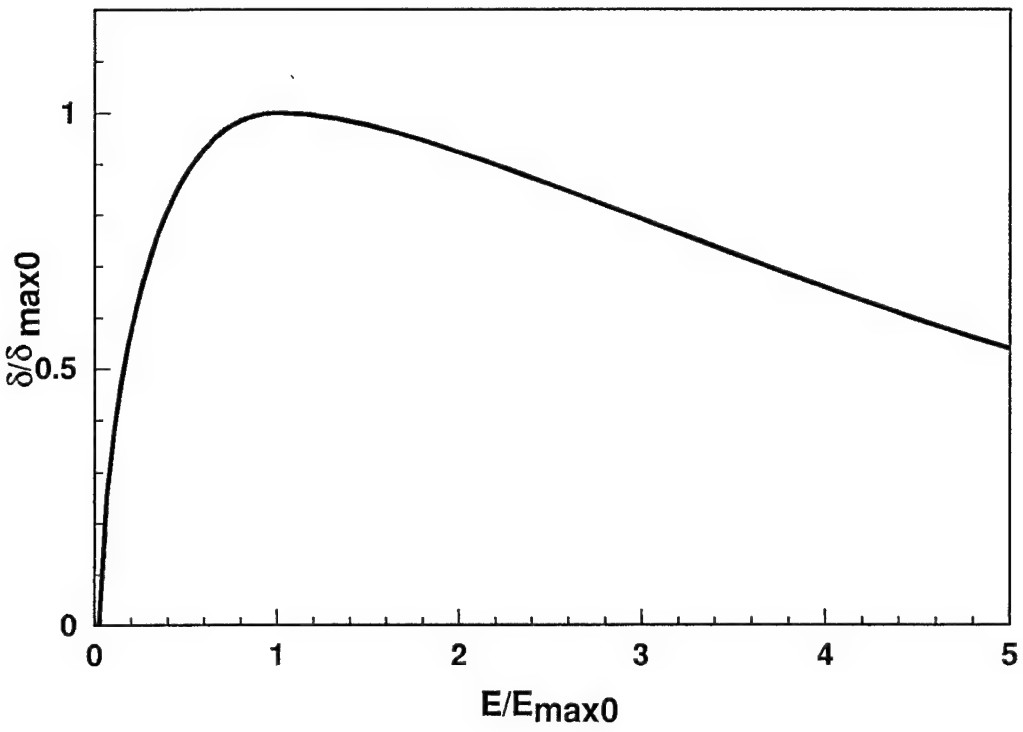


Figure 1: Normalized secondary emission coefficient as function of normalized energy at normal incidence, with $\mathcal{E}_{\max 0}/\mathcal{E}_0 = 40$, and $k_{sw} = 1$.

by:

$$w = \frac{\mathcal{E} - \mathcal{E}_0}{\mathcal{E}_{\max 0}\left(1 + k_{sw}\theta^2/2\pi\right) - \mathcal{E}_0}, \tag{2}$$

where $\mathcal{E}_0$ is the secondary emission threshold and k_{sw} is a surface-smoothness parameter similar to $k_{s\delta}$. Both $k_{s\delta}$ and k_{sw} vary between 0 for very rough surfaces and 2 for polished surfaces. Typical values are close to 1. The exponent k in Eq. 1 is given by:

$$k = \begin{cases} 0.62, & w < 1 \\ 0.25, & w \geq 1 \end{cases}. \tag{3}$$

The energy dependence of the secondary emission coefficient is shown in Fig. 1. The angular dependence of the secondary emission coefficient is shown in Fig. 2. Note that the secondary emission coefficient, which can be obtained by multiplying the energy dependent part by the angular dependent part, is largest at $\mathcal{E} = \mathcal{E}_{\max 0}$ and $\theta = 90$. The peak normal secondary emission in copper is $\delta_{\max 0} = 1.2$, with $\mathcal{E}_{\max 0} = 400$ eV, and $\mathcal{E}_0 = 15$.

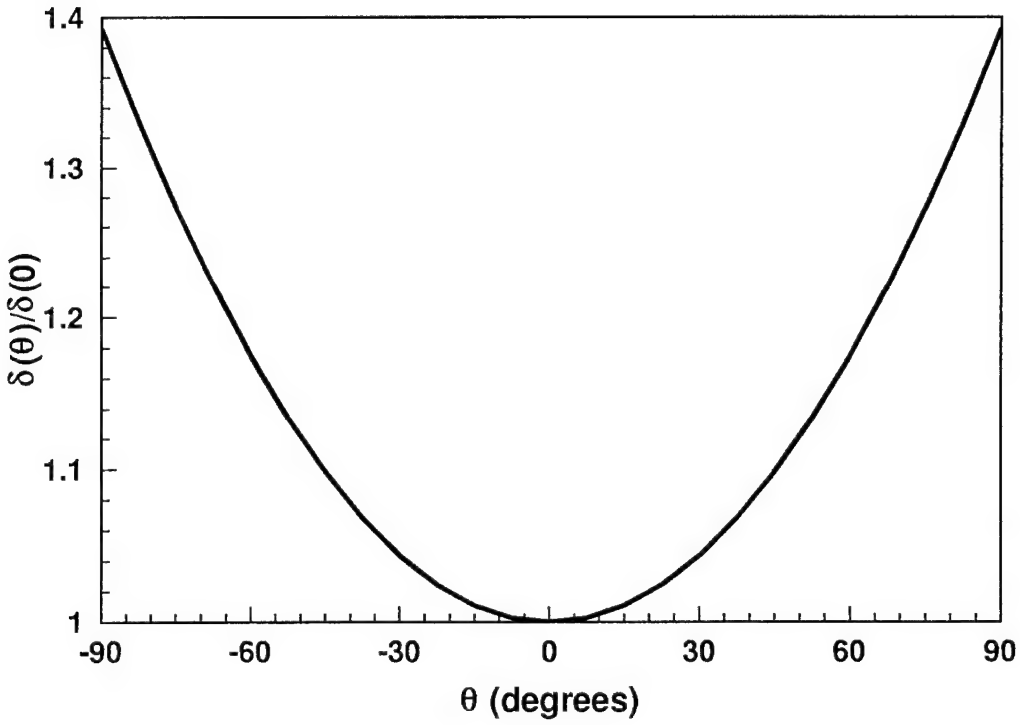


Figure 2: Angular dependence of the secondary emission coefficient normalized to the coefficient at normal incidence to the surface. Here $k_{s\delta} = 0$.

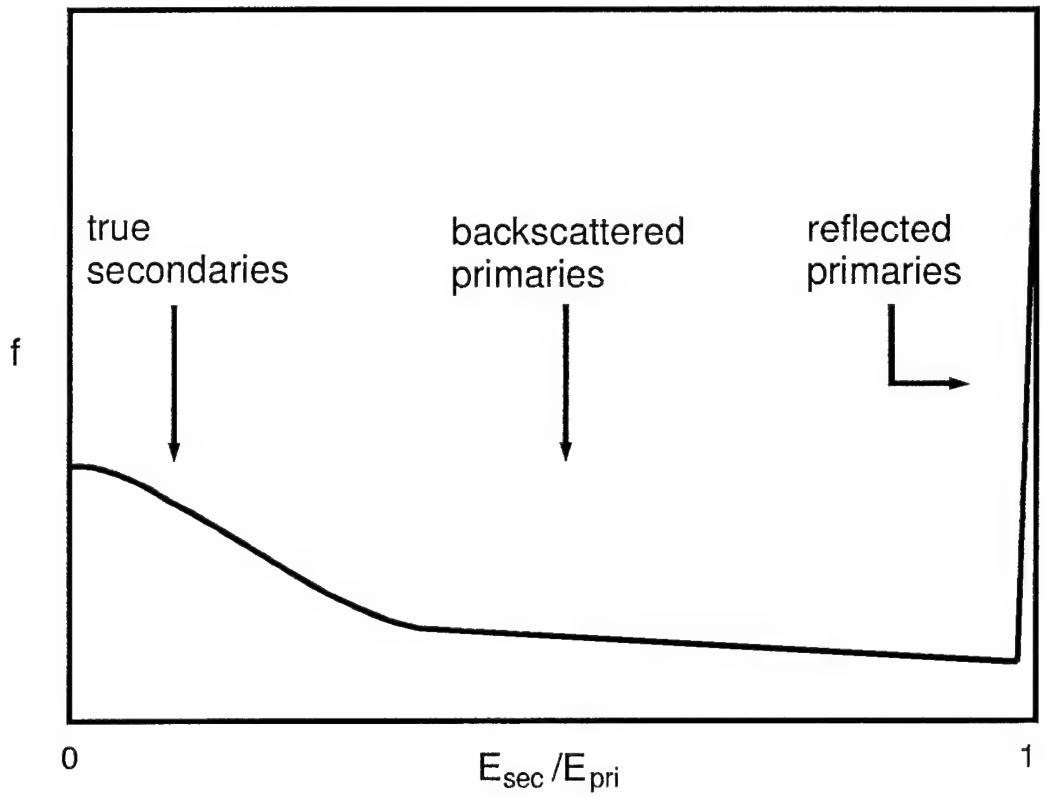


Figure 3: Schematic diagram of the secondary emission spectrum versus the ratio of the emitted energy of the secondary to the incident energy of the primary.

1.2 Secondary Emission Spectrum

The energy and angular distribution of the emitted secondaries were treated by Spangenberg [6], and are also treated by Vaughan [2]. The emission spectrum has three regimes, as shown in Fig.3. Incident electrons reflected at the surface of the secondary emitting material are called reflected primaries. Reflected primaries comprise about 3% of the emitted electron population. The energy of the reflected primary is approximately the same as the energy of the incident primary, $\mathcal{E}_r = \mathcal{E}_i$. The primary is reflected specularly in angle, $\theta_r = -\theta_i$, where θ is measured from the surface normal at the point of impact.

Backscattered primaries are electrons that impact the surface, and scatter off of several lattice atoms and/or impurities.Typically backscattered primaries comprise about 7% of the emitted electron population. These electrons reemerge from the impact surface with energies in the range $0 < \mathcal{E}_b < \mathcal{E}$. Within this energy range, all energies are taken as equally probable, $\mathcal{E}_b = R\mathcal{E}_i$, where

$0 \leq R < 1$ is a uniformly distributed random number. The angle of emission is taken to be specular, just as in the case of the reflected primaries. A more detailed treatment might consider a distribution of angles resulting from quantum mechanical treatment of scattering in the lattice potentials in the secondary emission medium.

True secondaries are electrons that are emitted from the conduction or valence bands of the atoms comprising the impact surface. The emitted electron population contains about 90% true secondaries. Energy is imparted to the lattice electrons over some time long compared to the elastic collision time, so the electron energy distribution can be modeled as a Maxwell-Boltzmann distribution. The electrons are emitted with energies from a thermal distribution of temperature T:

$$f(\mathcal{E}) = \frac{\mathcal{E}}{(k_B T)^2} \exp\left(\frac{-\mathcal{E}}{k_B T}\right), \tag{4}$$

where k_B is Boltzmann's constant. Due to the timescale of the emission process, the angle of emission can be taken to be isotropic:

$$g(\theta) = \frac{\cos(\theta)}{2}. \tag{5}$$

1.3 Discussion

The secondary model described above has been implemented in part in the XPDP1 code [7], and in full in the XOOPIC code [1]. In the scheme described above, the secondary emission coefficient can indicate that a fractional number of electrons must be emitted. The above codes emit fractional electron yields statistically, using a random number. Another technique is to accumulate fractional particles until a sufficient level is achieved to emit a particle. Variable particle weighting can also be used to represent fractional electron emission [8]. The secondary electrons are typically computed after each electron is absorbed at a surface, at the end of a timestep, and advanced into the simulation in the next timestep (XPDP1). Riyopoulus [9] has noted a banding effect when emitting at the end of a timestep, due to an average emission delay of $\Delta t/2$. This was repaired by computing a fractional step for emission which accounts for the time within the timestep that the particle impacted the emission surface. XOOPIC is capable of advancing each particle the remaining fraction of the timestep, eliminating this problem. The banding effect is observed only for large timesteps, $\omega_{rf}\Delta t/2 \approx 1$, where ω_{rf} is the driving frequency of the multipactor in the Riyopoulus study.

References

[1] J. P. Verboncoeur, A. B. Langdon, and N. T. Gladd, "An object-oriented electromagnetic PIC code", *Comput. Phys. Commun.* **87**, 199-211 (1995).

[2] J. R. M. Vaughan, "A New Formula for Secondary Emission Yield", *IEEE Trans.* **ED-36**, 1963-1967 (1989).

[3] J. R. M. Vaughan, "Secondary Emission Formulas", *IEEE Trans.* **ED-40**, 830 (1993).

[4] A. Shih and C. Hor, "Secondary Emission Properties as a Function of the Electron Incidence Angle", *IEEE Trans.* **ED-40**, 824-829.

[5] V. P. Gopinath, J. P. Verboncoeur and C. K. Birdsall, "Multipactor electron diascharge physics using an improved secondary emission model", *Phys. Plasmas* **5**, 1535-1540 (1998).

[6] K. R. Spangenberg, *Vacuum Tubes*, McGraw-Hill, New York, NY, pp. 48-57 (1948).

[7] J. P. Verboncoeur, M. V. Alves, V. Vahedi, and C. K. Birdsall, "Simultaneous Potential and Circuit Solution for 1d bounded particle simulation codes", *J. Comput. Phys.* **104**, 321-328 (1993).

[8] S. Riyopoulus, *Phys. Plasmas* **3**, 1481 (1997).

[9] S. Riyopoulus, D. Chernin, and D. Dialetis, *IEEE Tran.* **ED 44**, 489 (1997).

Digital filtering in radial coordinates for particle simulation

J. P. Verboncoeur
Dept. Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1770
johnv@eecs.berkeley.edu

to be submitted to *J. Comp. Phys.*

Abstract

A digital filtering algorithm for particle simulation in cylindrical coordinates is derived. For application of the filter to the charge density, the filter has the desirable property that total charge is conserved, and the filter preserves a uniform density. Analysis indicates the filter strength decreases with radius, and that hundreds of applications are required to reduce statistical fluctuations caused by finite size particles weighted to the mesh. A scheme is presented to apply n passes of the filter using a single matrix multiplication and a precomputed matrix. The filter is demonstrated on a high density lamp discharge plasma in cylindrical coordinates.

1 Introduction

Statistical noise has long been a significant limitation for particle-in-cell (PIC) calculations at high densities. This noise is manifested in the charge and current density source terms to the field equations, which results in fluctuations in fields. Such fluctuations in fields can cause non-physical net heating of particles [1]. The noise and consequent fluctuations are inversely proportional to the square root of the number of particles, so the brute force reduction of noise by increasing the number of particles is computationally inefficient. The problem is significantly more severe in curvilinear coordinates, where the particle density is proportional to the radius r in cylindrical coordinates, and proportional to r^2 in spherical coordinates.

A common remedy for this problem is the use of spectral filters or digital filters [1],[2]. Spectral filters operate in k-space, most appropriate when a spectral field solution is performed. Digital filters are more amenable to finite difference field solutions in real space, which is generally the case for bounded PIC codes in curvilinear coordinates.

The objective of this work is to extend digital filtering schemes to cylindrical coordinates. It is sufficient to consider the one-dimensional cylindrical case here, as the work can be extended to multiple dimensions in a straightforward fashion with the usual algebraic complications. In Section 2, the algorithm is derived and an implementation is described. In Section 3, a scheme for applying the equivalent of multiple passes of the filter with a single matrix-vector multiply using a pre-computed filter matrix is described. In Section 4, the algorithm is analyzed for both single and multiple passes, and the effect of the filter is demonstrated. In Section 5, the conclusions are presented.

2 Algorithm

We would like to consider a digital filter in cylindrical coordinates similar to the 1-2-1 spatial filter in Cartesian coordinates discussed in Birdsall and Langdon [1]. This type of filter is useful for reducing the short wavelength fluctuations generated by discrete particle shapes which can lead to particle heating. Note that the particle noise scales as the square root of the number of particles; a short wavelength filter can reduce the number of particles required to meet some noise-induced heating criterion in a particle simulation, relaxing demands on computing time to achieve the desired noise level.

Consider a general one-dimensional mesh, in which the volumes, V_j, are specified for $0 \leq j \leq N$. For physical reasons, we require conservation of charge, $Q^{n+1} = Q^n$, where the superscript refers to the iteration of smoothing, and

$$Q^n = \sum_{j=0}^{N} V_j \rho_j^n. \tag{1}$$

We must choose some other criterion for the coefficients of the filter. For a cylindrical system, it is possible that a Bessel function might be a more natural state, but here for simplicity we choose a function which will not perturb a uniform density in analogy to the planar case:

$$\begin{aligned} \rho_0^{n+1} &= (1-\alpha_0)\,\rho_0^n + \alpha_0 \rho_1^n, \\ \rho_j^{n+1} &= \alpha_j \rho_{j-1}^n + (1-2\alpha_j)\,\rho_j^n + \alpha_j \rho_{j+1}^n,\ 0 < j < N, \\ \rho_N^{n+1} &= \alpha_N \rho_{N-1}^n + (1-\alpha_N)\,\rho_N^n. \end{aligned} \tag{2}$$

Here, α_j are coefficients to be determined by satisfying charge conservation. We can see by inspection that a spatially uniform density will remain uniform after filtering by Eq. 2.

Applying charge conservation to Eq. 2, we obtain

$$\begin{aligned} \sum_{j=0}^{N} V_j \rho_j^n \;=\; \sum_{j=0}^{N} V_j \rho_j^{n+1} &= V_0\left[(1-\alpha_0)\,\rho_0^n + \alpha_0 \rho_1^n\right] + V_N\left[\alpha_N \rho_{N-1}^n + (1-\alpha_N)\,\rho_N^n\right] \\ &+ \sum_{j=1}^{N-1} V_j\left[\alpha_j\left(\rho_{j-1}^n + \rho_{j+1}^n\right) + (1-2\alpha_j)\,\rho_j^n\right]. \end{aligned} \tag{3}$$

Comparing like terms of ρ_j^n, we obtain a set of linear equations for the coefficients:

$$\begin{aligned} -V_0\alpha_0 + V_1\alpha_1 &= 0, \\ V_{j-1}\alpha_{j-1} - 2V_j\alpha_j + V_{j+1}\alpha_{j+1} &= 0, \\ V_{N-1}\alpha_{N-1} - V_N\alpha_N &= 0. \end{aligned} \tag{4}$$

However, these are not linearly independent, as consideration of the 3-row case demonstrates:

$$\begin{pmatrix} -V_0 & V_1 & 0 \\ V_0 & -2V_1 & V_2 \\ 0 & V_1 & -V_2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = 0. \tag{5}$$

Clearly the middle row is the opposite of the sum of the other two rows. This means we must obtain one additional equation. If we place a constraint on the filter that it should not change the sign of the contribution of a given charge density, ρ_j^n to the co-located filtered density, ρ_j^{n+1}, then we constrain $0 \leq \alpha_0 \leq 1$. So that this algorithm returns the familiar (and well-studied) 1-2-1

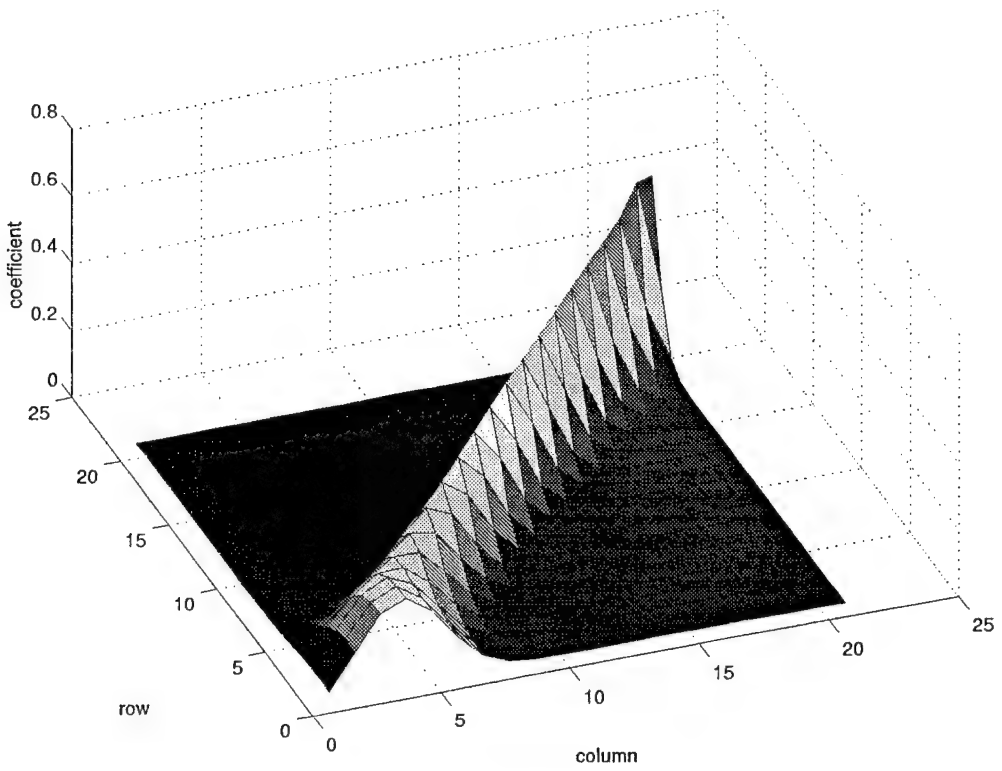


Figure 1: Multipass digital filter coefficients for a 100 pass filter on a 20 by 20 uniformly spaced radial mesh in cylindrical coordinates, with $\alpha_0 = 1/2$. The filter coefficients for a fixed row (horizontal slice) show the contribution of the unfiltered data to a particular filtered element, while the coefficients for a fixed column (vertical slice) show the redistribution profile for a unit charge at the appropriate radius.

digital filter for uniform meshes in Cartesian coordinates, we might choose $\alpha_0 = 1/2$. This results in the generating sequence for the coefficients:

$$\begin{gathered} \alpha_0 = 1/2, \\ \alpha_1 = \alpha_0 V_0/V_1, \\ \alpha_j = 2\alpha_{j-1}V_{j-1}/V_j - \alpha_{j-2}V_{j-2}/V_j,\ 2 \le j < N, \\ \alpha_N = \alpha_{N-1}V_{N-1}/V_N. \end{gathered} \tag{6}$$

The recursion given in Eq. 6 yields the familiar 1-2-1 digital filter when the volumes are equal (except the end cells, which have half the volume of the center cells). It conserves charge for arbitrary volumes, and preserves a uniform distribution for arbitrary volumes. This algorithm should work for both uniform and non-uniform meshes in any coordinate system in one dimension.

3 Multiple Applications of the Digital Filter

As shown above, the particle shape is weakly impacted by the smoothing in cylindrical coordinates. Empirical evidence also indicates that hundreds of passes of smoothing are required to reduce noise in discharges in order to reduce numerical heating. Since the algorithm can become expensive with

hundreds of passes of Eq. 2, we consider a scheme by which n passes of the filter can be applied simultaneously. Taking the filter to be of the form

$$\rho^{n+1} = A\rho^n, \tag{7}$$

where ρ^n is the charge density vector over the spatial dimension after n applications of the filter, and A is the matrix constructed from α_j given by

$$A = \begin{pmatrix} 1-\alpha_0 & \alpha_0 & 0 & ... & \\ \alpha_1 & 1-2\alpha_1 & \alpha_1 & & ... \\ 0 & & ... & & 0 \\ ... & & \alpha_{N-1} & 1-2\alpha_{N-1} & \alpha_{N-1} \\ & ... & 0 & \alpha_N & 1-\alpha_N \end{pmatrix}. \tag{8}$$

When we apply multiple passes of the digital filter, we simply nest applications of the linear operation given in Eq. 7. This leads to the easily implemented and efficient multiple pass algorithm,

$$\rho^n = A^n\rho^0, \tag{9}$$

where A^n can be precomputed for applications where the digital filter level is prescribed initially.

The coefficient matrix for a 20 by 20 uniformly spaced mesh, with the initial coefficient $\alpha_0 = 1/2$, is shown in Fig. 1. Slices with fixed row numbers characterize the radial spread of the filtered quantity at a given radius, from which we see that the filtering produces broadening of an initial distribution which increases with decreasing radius. Slices with fixed column numbers show the contributions of the pre-filtered components to a component at the radius corresponding to the column.

4 Discussion

In planar coordinates, the shape of a linearly weighted charge in one dimension remains linear, but the width of the charge increases in proportion to the decrease in the peak. In cylindrical coordinates, the analogous weighting, found from recursing Eq. 6 with $\alpha_0 = 1/2$, results in a much milder impact on the particle shape. The particle shapes for the initial linear spline and 1, 10, and 100 applications of the $\alpha_0 = 1/2$ filter are shown in Fig. 2. A single application of the filter is almost indistinguishable from the original linear particle shape. After 10 applications a small change in shape is discernible, and after 100 applications the shape is considerably smoother. Note that the filtered shapes have a discontinuity in the first derivative at each cell edge, and the second derivative on the left side of the particle is slightly positive, while it is slightly negative on the right side.

Spectral analysis corresponding to the above figure.

The multi-pass digital filter has been applied to a high density 1 cm cylindrical discharge using the code XPDC1. The discharge is a model of a fluorescent lamp discharge, which suffers from significant numerical heating due to the statistical fluctuations which are approximately proportional to $1/r$. This unfavorable radial scaling makes the heating problem very expensive to address by increasing the number of particles, N_p, since the noise level is proportional to $N_p^{-1/2}$. The charge density from the multi-pass digital filter with $n = 400$ is compared to the initial charge density obtained from linear weighting in Fig. 3. Note the $1/r$ characteristic envelope on the unfiltered charge density, and the reduction in short wavelength fluctuations with the filter.

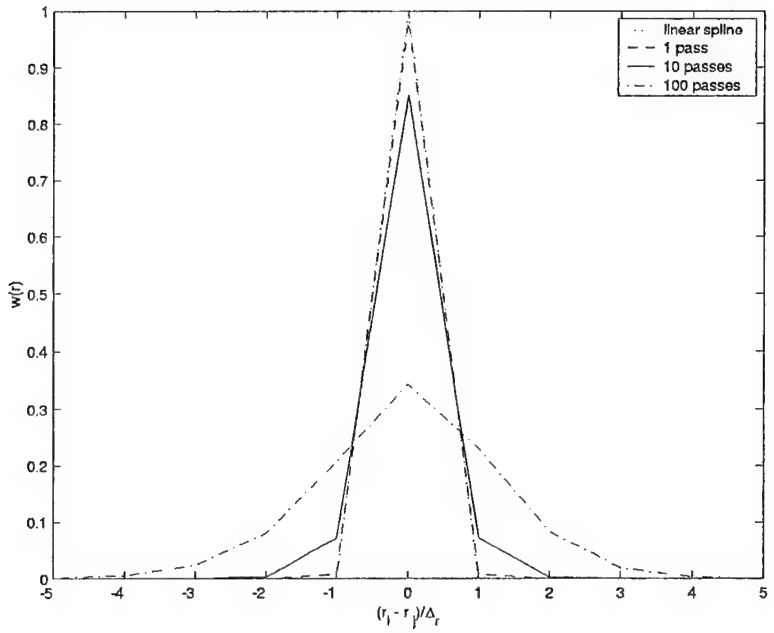


Figure 2: Shape of a standard linearly weighted particle, and shape after 1, 10, and 100 passes of the $\alpha_0 = 1/2$ cylindrical filter. The linear spline and 1-pass curve are nearly co-located.

5 Conclusions

A digital filter for discrete functions such as charge density was derived for cylindrical coordinates. The filter is designed to have the desirable characteristics of charge conservation, as well as preserving a uniform density. Although single passes of the filter have only a small effect, a scheme is described which can apply the equivalent of an arbitrary number of filter passes with a single matrix-vector multiplication using a sparse matrix computed initially. The digital filter has a small per-pass effect, but can have a significant multi-pass effect. The digital filter is demonstrated to be effective in reducing statistical fluctuations in the charge density for a high density lamp discharge which would suffer from numerical heating without the filter.

6 Acknowledgments

This work was supported in part by General Electric Corporation contract GE-20000181-BIRDSALL-02/02 and AFOSR STTR F49620-99-C-0028.

References

[1] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* McGraw-Hill, New York N.Y., 1985.

[2] R. W. Hamming. *Numerical Methods for Scientists and Engineers.* McGraw-Hill, New York, N.Y., 1973.

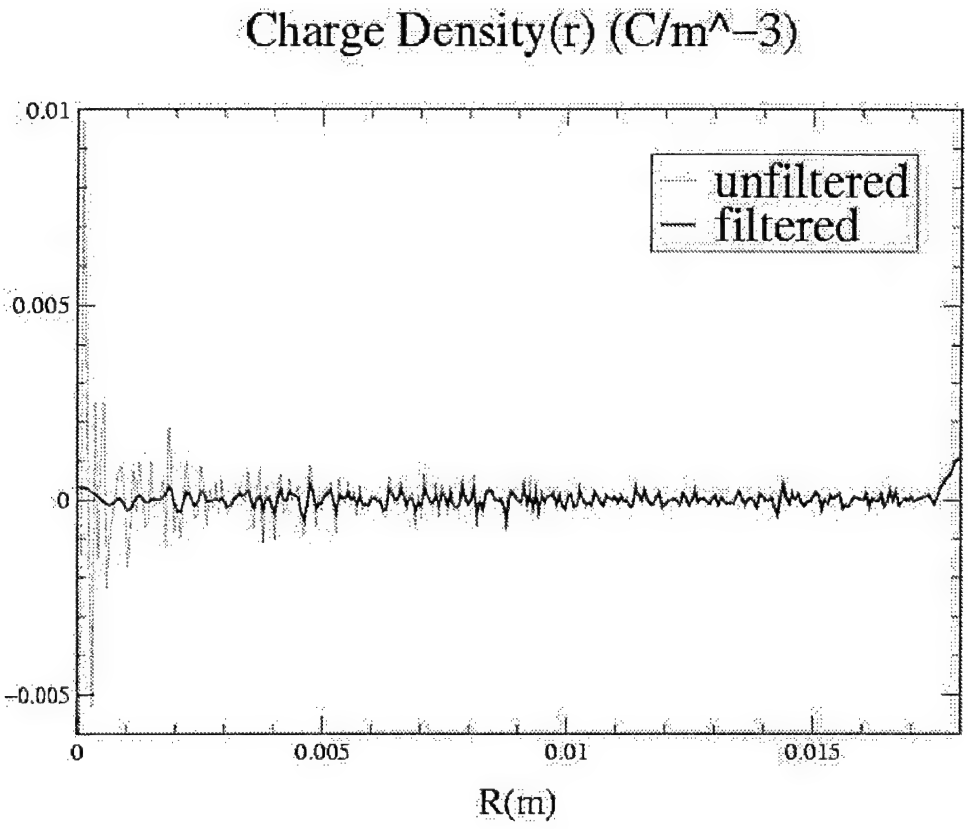


Figure 3: Charge density from an $n = 400$ multipass filter compared to unfiltered data. The discharge is net neutral except in the sheath, so all short wavelength fluctuations inside the sheath radius are attributed to statistical noise.

Symmetric spline weighting for charge and current density in particle simulation

J. P. Verboncoeur

Dept. EECS, University of California, Berkeley, CA 94720-1770

Abstract

A general method for computing charge and current density source terms for Maxwell's equations from particles weighted to a mesh is described. The method presented here eliminates the need for correction factors often applied in curvilinear coordinates to compensate for errors at the edge of the system, and in the interior as well for nonuniform meshes. Generality is achieved by weighting volume elements using a spline symmetric to that by which particle charge and current are weighted to the mesh. The method presented has a number of desirable properties, including conservation of charge, preservation of a uniform distribution, and generality on nonuniform meshes with arbitrary particle-mesh interpolation schemes. The method recovers the exact answer in the limit of mesh sizes approaching zero.

Keywords: PIC, particle, weighting, radial, cylindrical, spherical, correction

1 Introduction

In charge and current accumulation schemes commonly used in particle-in-cell codes, a systematic error occurs on the boundary cells in curvilinear coordinates. The error is particularly severe on the axis in cylindrical and spherical models, with systematic errors of 33% and 100% larger than the theoretical value for either charge or current density in cylindrical and spherical coordinates, respectively. Additional error occurs at the outer edge of the system, and throughout the interior for non-uniform meshes. This error leads to similar errors in the forces calculated at those points, as well as exacerbating radial noise due to particle statistics, which is largest near $r = 0$. The errors in density do *not* decrease with decreasing mesh size. This problem is not present in Cartesian coordinates, even for non-uniform meshes.

The issue of radial correction factors for the charge density term has been addressed in recent works by Ruyten [1] and by Larsen et Al.. [2]. These works derive the error for linear weighting to uniform meshes in cylindrical coordinates and propose correction terms. In [1], the author states that the axial behavior is beyond the scope of the work. In [2], the analysis is carried out including the origin, and the proposed correction is shown to approach the theoretical value for large numbers of particles. We will demonstrate that the result of [2] is exact for the uniform charge density case despite errors shown in that paper for charge density at the axis.

In this work, the error is analyzed for the uniform particle density distribution for the charge density, ρ. A general scheme is developed which applies the same interpolation scheme to both the particle and the volume or surface element in order to obtain a density which eliminates the systematic error in curvilinear coordinates. The new scheme is compared with analytic theory in cylindrical and spherical coordinates. The algorithm is extended to the current density, J.

2 Density Errors

In this section, the errors in the most common interpolation scheme used in particle simulation [3], linear weighting, are demonstrated for uniform density in one dimensional on a nonuniform mesh in cylindrical and spherical coordinates. For an arbitrary continuum particle distribution specified by $f(r)$, the density in cylindrical coordinates is given by:

$$n(r) = \frac{\int_r^{r+dr} f(r')dr'}{\int_r^{r+dr} 2\pi r' dr'}. \tag{1}$$

For the uniform particle distribution, $f(r) = 2\pi r$, and we obtain $n(r) = 1$. Similarly, in spherical coordinates for the uniform particle distribution $f(r) = 4\pi r^2$ we obtain:

$$n\left(r\right) = \frac{\int_r^{r+dr} f(r')dr'}{\int_r^{r+dr} 4\pi \left(r'\right)^2 dr'} = 1 \tag{2}$$

For linear weighting on a nonuniform mesh in cylindrical coordinates, we can write the density at an arbitrary intermediate node for an arbitrary particle distribution $f(r)$ in the standard way [3]:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1/2}}^{r_{j+1/2}} 2\pi r dr}, \tag{3}$$

where the r_j refer to the position of the jth mesh, and $r_{j+1/2} \equiv \left(r_j + r_{j+1}\right)/2$. The edge densities are written (valid also on axis, $r_0 = 0$):

$$n_0 = \frac{\int_{r_0}^{r_1} f(r)\frac{r_1-r}{r_1-r_0}dr}{\int_{r_0}^{r_{1/2}} 2\pi r dr}, \text{ and} \tag{4}$$

$$n_N = \frac{\int_{r_{N-1}}^{r_N} f(r)\frac{r-r_{N-1}}{r_N-r_{N-1}}dr}{\int_{r_{N-1/2}}^{r_N} 2\pi r dr}. \tag{5}$$

For a uniform particle distribution, $f(r) = 2\pi r$, the standard discrete densities in Eqs. 3-5 become:

$$n_j = \frac{4}{3}\frac{r_{j-1}+r_j+r_{j+1}}{r_{j-1}+2r_j+r_{j+1}}, \tag{6}$$

$$n_0 = \frac{4}{3}\frac{2r_0+r_1}{3r_0+r_1}, \text{ and} \tag{7}$$

$$n_N = \frac{4}{3}\frac{r_{N-1}+2r_N}{r_{N-1}+3r_N}. \tag{8}$$

On a non-uniform mesh, the linear weighting with uncorrected volumes always produces the incorrect result for all cells. For a uniform mesh, we see from inspection of Eq. 6 that the correct solution is produced for the interior cells, but the systematic error persists at the edges. For nonuniform meshes, the error occurs at interior points as well. The systematic error is 1/3 on axis for meshes which include the axis, and is independent of grid spacing Δ_r. Note that the error in the outer edge is small for $N \gg 1$. The uncorrected results for a uniform and nonuniform mesh are plotted in Fig. 1.

A similar development for linear weighting in spherical coordinates for the uniform particle distribution, $f(r) = 4\pi r^2$, gives:

$$n_j = 2\frac{r_{j-1}^2+r_{j-1}r_j+r_{j-1}r_{j+1}+r_j^2+r_jr_{j+1}+r_{j+1}^2}{r_{j-1}^2+3r_{j-1}r_j+r_{j-1}r_{j+1}+3r_j^2+3r_jr_{j+1}+r_{j+1}^2}, \tag{9}$$

$$n_0 = 2\frac{3r_0^2+2r_0r_1+r_1^2}{7r_0^2+4r_0r_1+r_1^2}, \text{ and} \tag{10}$$

$$n_N = 2\frac{r_{N-1}^2+2r_{N-1}r_N+3r_N^2}{r_{N-1}^2+4r_{N-1}r_N+7r_N^2}. \tag{11}$$

For uniform mesh spacing, Δ_r, Eqs. 9-11 become:

$$n_j = 2\frac{6r_j^2+\Delta_r^2}{12r_j^2+\Delta_r^2}, \tag{12}$$

$$n_0 = 2\frac{6r_0^2+4r_0\Delta_r+\Delta_r^2}{12r_0^2+6r_0\Delta_r+\Delta_r^2}, \text{ and} \tag{13}$$

$$n_N = 2\frac{6r_N^2-4r_N\Delta_r+\Delta_r^2}{12r_N^2-6r_N\Delta_r+\Delta_r^2}. \tag{14}$$

Note that the uncorrected density for the spherical case is always in error, approaching the correct solution only for $r_j \gg \Delta_r$. The uncorrected results for a uniform and nonuniform mesh are plotted in Fig. 2. The systematic error is 100% on axis for meshes which include the axis, independent of grid spacing Δ_r.

3 Improved Density Weighting Algorithm

In this section, a general algorithm is developed for computing the charge density in non-uniform meshes for curvilinear coordinates using arbitrary particle-mesh interpolation. Consider a modified calculation for the volume in which differential volume elements are weighted to the mesh using the same algorithm as the charge weighting. A general method is suggested by the definition of density in cylindrical (Eq. 1) and spherical (Eq. 2) coordinates, weighting both the charge and the volume:

$$n_{\mathbf{j}} = \frac{\int_{\mathbf{r}} f(r) W_{\mathbf{j}}(\mathbf{r})\, dr}{\int_{\mathbf{r}} W_{\mathbf{j}}(\mathbf{r})\, dV}, \tag{15}$$

where $W_{\mathbf{j}}(\mathbf{r})$ is an interpolation function which weights particles at position $\mathbf{r}$ to mesh $\mathbf{j}$, and dV is a volume element, given in one dimension by $dV = 2\pi r$ in cylindrical coordinates and $dV = 4\pi r^2$ in spherical coordinates.

For linear weighting in cylindrical coordinates, the density can then be written:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1}}^{r_j} 2\pi r\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} 2\pi r\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}. \tag{16}$$

The edge densities are found simply by dropping the out of bounds integrals:

$$n_0 = \frac{\int_{r_0}^{r_1} f(r)\frac{r_1-r}{r_1-r_0}dr}{\int_{r_0}^{r_1} 2\pi r\frac{r_1-r}{r_1-r_0}dr}, \tag{17}$$

and

$$n_N = \frac{\int_{r_{N-1}}^{r_N} f(r)\frac{r-r_{N-1}}{r_N-r_{N-1}}dr}{\int_{r_{N-1}}^{r_N} 2\pi r\frac{r-r_{N-1}}{r_N-r_{N-1}}dr} \tag{18}$$

For $f(r) = 2\pi r$, we obtain the exact solution for all $0 \le j \le N$, $n_j = 1$. For more general particle distributions (specifically when f is not a linear function of r), Eqs. 16-18 result in an error term which depends upon the weighting scheme; for example the linear weighting function results in a error proportional to Δ_r^2.

This can be implemented in a standard linear weighting scheme simply by modifying the volumes used to compute n_j. Taking the cell size in the z-direction to be Δ_z, the volumes become:

$$V_j = \Delta_z \frac{\pi}{3}\left[r_{j+1}\left(r_j + r_{j+1}\right) - r_{j-1}\left(r_{j-1} + r_j\right)\right], \tag{19}$$

$$V_0 = \Delta_z \frac{\pi}{3}\left(r_1 - r_0\right)\left(2r_0 + r_1\right), \text{ and} \tag{20}$$

$$V_N = \Delta_z \frac{\pi}{3}\left(r_N - r_{N-1}\right)\left(r_{N-1} + 2r_N\right). \tag{21}$$

The corrected and uncorrected densities have been implemented in the XOOPIC [4] and XPDC1 [5] codes, and comparisons of the results are shown in Fig. 1. The comparisons shown are valid when there is at least 1 particle per cell. Total charge and volume are also identically conserved with the corrected method. This method can be easily extended to arbitrary weighting functions by using the desired weighting in Eqs. 16-18.

For linear weighting in spherical coordinates the density can be written:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1}}^{r_j} 4\pi r^2\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} 4\pi r^2\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}, \tag{22}$$

with edge densities again obtained by dropping out of bounds integrals.

For the uniform particle distribution, $f(r) = 4\pi r^2$, we again obtain the exact solution for all $0 \le j \le N$, $n_j = 1$. For more general particle distributions we obtain the same result as that of the cylindrical case above.

Similar to the cylindrical case, the method can be implemented for the spherical scheme by precomputing the volumes using the denominator of Eq. 22:

$$V_j = \frac{\pi}{3}(r_{j+1} - r_{j-1})\left(r_{j-1}^2 + r_{j-1}r_j + r_{j-1}r_{j+1} + r_j^2 + r_j r_{j+1} + r_{j+1}^2\right), \tag{23}$$

$$V_0 = \frac{\pi}{3}(r_1 - r_0)\left(3r_0^2 + 2r_1 r_0 + r_1^2\right), \text{ and} \tag{24}$$

$$V_N = \frac{\pi}{3}(r_N - r_{N-1})\left(r_{N-1}^2 + 2r_{N-1}r_N + 3r_N^2\right). \tag{25}$$

The densities computed with the method presented here are compared to the uncorrected densities for linear weighting on a uniform and nonuniform mesh in Fig. 2.

4 Current Density

This algorithm can be applied to current density for the electromagnetic source term in a straightforward manner by using $J = nqv$. The quantity qv is weighted to the mesh for each particle, and the result is divided by the surface area, $\mathbf{S}$. To obtain the surface area, consider the differential elements of the components in cylindrical coordinates,

$$dS_r = rd\theta dz, \tag{26}$$

$$dS_\theta = drdz, \text{ and} \tag{27}$$

$$dS_z = rdrd\theta. \tag{28}$$

It is evident from Eqs. 26-28 that only the axial component of **S** will result in a nonlinear dependence on r when computing the surface area from $S = \int W dS$, where W is the particle-mesh weighting used. The results for the axial component of the current density are trivially different from the charge density results derived above, and the comparison plots are identical.

5 Conclusions

An algorithm for obtaining the correct charge and current densities in curvilinear coordinate systems for arbitrary particle interpolation schemes is described. Volumes and surface areas are weighted to the mesh using the same interpolation scheme used to weight particles. The method recovers the charge density correction factors for linear weighting in cylindrical coordinates [2], but extends to the more general case. The method has the notable beneficial properties of conserving charge and current as well as total volume and surface area on a general orthogonal mesh. The algorithm yields the exact answer for the uniform particle distribution, as demonstrated here; arbitrary distributions are similarly correct to the mesh resolution, becoming exact as the mesh size approaches zero for continuum distributions. This is a significant departure of the pervious scheme of computing using incorrect volumes and then using correction factors designed for special cases [2]. The results are easily extended to two and three dimensions.

6 Acknowledgments

This work supported in part by Department of Energy SBIR contract DE-FG03-99ER82903, and General Electric Corporation contract GE-20000181-BIRDSALL-02/02. The author is grateful for useful discussions with H. B. Smith, H. J. Lee, K. L. Cartwright, A. B. Langdon, and C. K. Birdsall.

References

[1] W. M. Ruyten. Density-conserving shape factors for particle simulation in cylindrical and spherical coordinates. *Journal of Computational Physics*, 105:224, 1993.

[2] D. J. Larson, D. W. Hewett, and A. B. Langdon. Correction factors for PIC accumulation on radial grids. *Computer Physics Communications*, 90:260–266, 1995.

[3] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* McGraw-Hill, New York N.Y., 1985.

[4] J. P. Verboncoeur, A. B. Langdon, and N. T. Gladd. An object-oriented electromagnetic PIC code. *Comp. Phys. Comm.*, 87:199–211, 1995.

[5] J. P. Verboncoeur, M. V. Alves, V. Vahedi, and C. K. Birdsall. Simultaneous potential and circuit solution for 1d bounded plasma particle simulation codes. *J. Comp. Phys.*, 104:321–328, 1993.

List of Figures

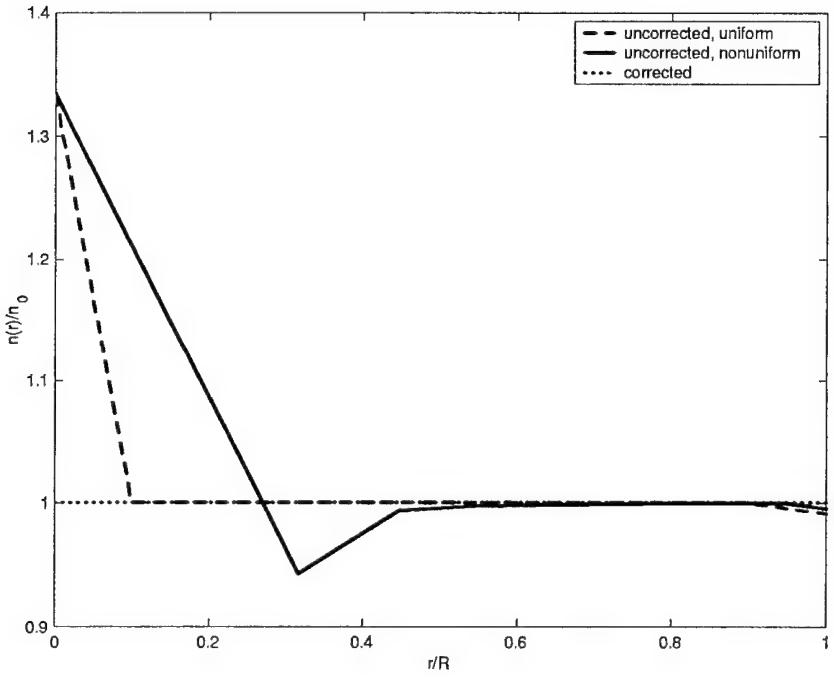


Figure 1: Mesh densities computed using standard linear weighting (uncorrected) on a uniform and nonuniform mesh in cylindrical coordinates, as well as the corrected density on both meshes. The uniform mesh has $N = 10$ steps; the nonuniform mesh is computed from $r_j = \sqrt{r_{j-1}^2 + \left(r_N^2 - r_0^2\right)/N}$, such that the interval contains $N = 10$ equal volumes.

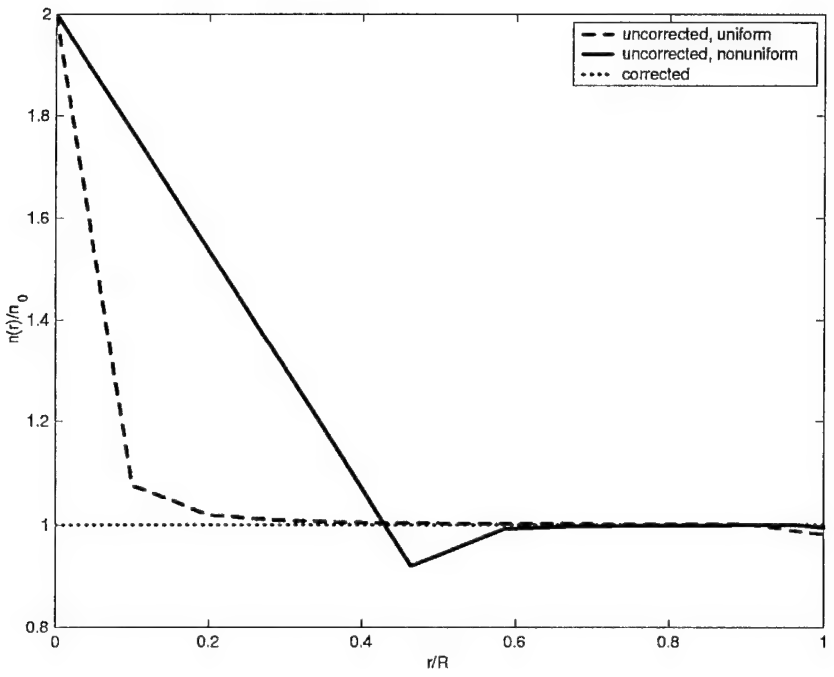


Figure 2: Mesh densities computed using standard linear weighting (uncorrected) on a uniform and nonuniform mesh in spherical coordinates, as well as the corrected density on both meshes. The uniform mesh has $N = 10$ steps; the nonuniform mesh is computed from $r_j = \left[r_{j-1}^3 + \left(r_N^3 - r_0^3\right)/N\right]^{1/3}$, such that the interval contains $N = 10$ equal volumes.

Symmetric spline weighting for charge and current density in particle simulation

J. P. Verboncoeur

Dept. Nuclear Engineering, University of California, Berkeley, CA 94720-1770

E-mail: johnv@eecs.berkeley.edu

Communicated by Someone

Received 1 January 1999; revised 2 January 1999; accepted 3 January 1999

A general method for computing charge and current density source terms for Maxwell's equations from particles weighted to a mesh is described. The method presented here eliminates the need for correction factors often applied in curvilinear coordinates to compensate for errors at the edge of the system, and in the interior as well for nonuniform meshes. Generality is achieved by weighting volume elements using a spline symmetric to that by which particle charge and current are weighted to the mesh. The method presented has a number of desirable properties, including conservation of charge, preservation of a uniform distribution, and generality on nonuniform meshes with arbitrary particle-mesh interpolation schemes. The method recovers the exact answer in the limit of mesh sizes approaching zero.

Key Words: PIC, particle, weighting, radial, cylindrical, spherical, correction

CONTENTS

1. INTRODUCTION

In charge and current accumulation schemes commonly used in particle-in-cell codes, a systematic error occurs on the boundary cells in curvilinear coordinates. The error is particularly severe on the axis in cylindrical and spherical models, with systematic errors of 33% and 100% larger than the theoretical value for either charge or current density in cylindrical and spherical coordinates, respectively. Additional error occurs at the outer edge of the system, and throughout the interior for non-uniform meshes. This error leads to similar errors in the forces calculated at those points, as well as exacerbating radial noise due to particle statistics, which is largest near $r = 0$. The errors in density do *not* decrease with decreasing mesh size. This problem is not present in Cartesian coordinates, even for non-uniform meshes.

The issue of radial correction factors for the charge density term has been addressed in recent works by Ruyten [1] and by Larsen et al. [2]. These works derive the error for linear weighting to uniform meshes in cylindrical coordinates and propose correction terms. In [1], the author states that the axial behavior is beyond the scope of the work. In [2], the analysis is carried out including the origin, and the proposed correction is shown to approach the theoretical value for large numbers of particles. We will demonstrate that the result of [2] is exact for the uniform charge density case despite errors shown in that paper for charge density at the axis.

In this work, the error is analyzed for the uniform particle density distribution for the charge density, ρ. A general scheme is developed which applies the same interpolation scheme to both the particle and the volume or surface element in order to obtain a density which eliminates the systematic error in curvilinear coordinates. The new scheme is compared with analytic theory in cylindrical and spherical coordinates. The algorithm is extended to the current density, J.

2. DENSITY ERRORS

In this section, the errors in the most common interpolation scheme used in particle simulation [3], linear weighting, are demonstrated for uniform density in one dimensional on a nonuniform mesh in cylindrical and spherical coordinates. For an arbitrary continuum particle distribution specified by $f(r)$, the exact density in cylindrical coordinates is given by:

$$n(r) = \frac{\int_r^{r+dr} f(r')dr'}{\int_r^{r+dr} 2\pi r' dr'}. \tag{1}$$

For the uniform particle distribution, $f(r) = 2\pi r$, and we obtain $n(r) = 1$. Similarly, in spherical coordinates for the uniform particle distribution $f(r) = 4\pi r^2$ we obtain:

$$n\left(r\right) = \frac{\int_r^{r+dr} f(r')dr'}{\int_r^{r+dr} 4\pi \left(r'\right)^2 dr'} = 1 \tag{2}$$

In the classical particle scheme [3], the cell volumes are computed from geometric considerations independently of the weighting scheme used to accumulate charge to the grid. Because the volume elements are nonlinear in the spatial variable for curvilinear coordinates, this leads to systematic errors. This conceptual error is precisely why the previous works ([1] and [2]) describe corrections to the weighting scheme. For linear weighting on a nonuniform mesh in cylindrical coordinates, we can write the density at an arbitrary intermediate node for an arbitrary particle distribution $f(r)$ in the standard way [3] in order to make the conceptual error more apparent:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1/2}}^{r_{j+1/2}} 2\pi r dr}, \tag{3}$$

where the r_j refer to the position of the jth mesh, and $r_{j+1/2} \equiv \left(r_j + r_{j+1}\right)/2$. The edge densities are written (valid also on axis, $r_0 = 0$):

$$n_0 = \frac{\int_{r_0}^{r_1} f(r)\frac{r_1-r}{r_1-r_0}dr}{\int_{r_0}^{r_{1/2}} 2\pi r dr}, \text{ and} \tag{4}$$

$$n_N = \frac{\int_{r_{N-1}}^{r_N} f(r)\frac{r-r_{N-1}}{r_N-r_{N-1}}dr}{\int_{r_{N-1/2}}^{r_N} 2\pi r dr}. \tag{5}$$

For a uniform particle distribution, $f(r) = 2\pi r$, the standard discrete densities in Eqs. 3-5 become:

$$n_j = \frac{4}{3}\frac{r_{j-1}+r_j+r_{j+1}}{r_{j-1}+2r_j+r_{j+1}}, \tag{6}$$

$$n_0 = \frac{4}{3}\frac{2r_0+r_1}{3r_0+r_1}, \text{ and} \tag{7}$$

$$n_N = \frac{4}{3}\frac{r_{N-1}+2r_N}{r_{N-1}+3r_N}. \tag{8}$$

On a non-uniform mesh, the linear weighting with uncorrected volumes always produces the incorrect result for all cells. For a uniform mesh, we see from inspection of Eq. 6 that the correct solution is produced for the interior cells, but the systematic error persists at the edges. For nonuniform meshes, the error occurs at interior points as well. The systematic error is 1/3 on axis for meshes which include the axis, and is independent of grid spacing Δ_r. Note that the error in the outer edge is small for $N \gg 1$. The uncorrected results for a uniform and nonuniform mesh are plotted in Fig. 1.

A similar development for linear weighting in spherical coordinates for the uniform particle distribution, $f(r) = 4\pi r^2$, gives:

$$n_j = 2\frac{r_{j-1}^2+r_{j-1}r_j+r_{j-1}r_{j+1}+r_j^2+r_jr_{j+1}+r_{j+1}^2}{r_{j-1}^2+3r_{j-1}r_j+r_{j-1}r_{j+1}+3r_j^2+3r_jr_{j+1}+r_{j+1}^2}, \tag{9}$$

$$n_0 = 2\frac{3r_0^2+2r_0r_1+r_1^2}{7r_0^2+4r_0r_1+r_1^2}, \text{ and} \tag{10}$$

$$n_N = 2\frac{r_{N-1}^2+2r_{N-1}r_N+3r_N^2}{r_{N-1}^2+4r_{N-1}r_N+7r_N^2}. \tag{11}$$

For uniform mesh spacing, Δ_r, Eqs. 9-11 become:

$$n_j = 2\frac{6r_j^2+\Delta_r^2}{12r_j^2+\Delta_r^2}, \tag{12}$$

$$n_0 = 2\frac{6r_0^2 + 4r_0\Delta_r + \Delta_r^2}{12r_0^2 + 6r_0\Delta_r + \Delta_r^2}, \text{ and} \tag{13}$$

$$n_N = 2\frac{6r_N^2 - 4r_N\Delta_r + \Delta_r^2}{12r_N^2 - 6r_N\Delta_r + \Delta_r^2}. \tag{14}$$

Note that the uncorrected density for the spherical case is always in error, approaching the correct solution only for $r_j \gg \Delta_r$. The uncorrected results for a uniform and nonuniform mesh are plotted in Fig. 2. The systematic error is 100% on axis for meshes which include the axis, independent of grid spacing Δ_r.

3. IMPROVED DENSITY WEIGHTING ALGORITHM

In this section, a general algorithm is developed for computing the charge density in non-uniform meshes for curvilinear coordinates using arbitrary particle-mesh interpolation. Consider a modified calculation for the volume in which differential volume elements are weighted to the mesh using the same algorithm as the charge weighting. A general method is suggested by the definition of density in cylindrical (Eq. 1) and spherical (Eq. 2) coordinates, weighting both the charge and the volume:

$$n_{\mathrm{j}} = \frac{\int_{\mathrm{r}} f(r) W_{\mathrm{j}}(\mathbf{r})\, dr}{\int_{\mathrm{r}} W_{\mathrm{j}}(\mathbf{r})\, dV}, \tag{15}$$

where $W_{\mathrm{j}}(\mathbf{r})$ is an interpolation function which weights particles at position $\mathbf{r}$ to mesh $\mathbf{j}$, and dV is a volume element, given in one dimension by $dV = 2\pi r$ in cylindrical coordinates and $dV = 4\pi r^2$ in spherical coordinates.

For linear weighting in cylindrical coordinates, the density can then be written:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1}}^{r_j} 2\pi r\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} 2\pi r\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}. \tag{16}$$

The edge densities are found simply by dropping the out of bounds integrals:

$$n_0 = \frac{\int_{r_0}^{r_1} f(r)\frac{r_1-r}{r_1-r_0}dr}{\int_{r_0}^{r_1} 2\pi r\frac{r_1-r}{r_1-r_0}dr}, \tag{17}$$

and

$$n_N = \frac{\int_{r_{N-1}}^{r_N} f(r)\frac{r-r_{N-1}}{r_N-r_{N-1}}dr}{\int_{r_{N-1}}^{r_N} 2\pi r\frac{r-r_{N-1}}{r_N-r_{N-1}}dr} \tag{18}$$

For $f(r) = 2\pi r$, we obtain the exact solution for all $0 \leq j \leq N$, $n_j = 1$. For more general particle distributions (specifically when f is not a linear function of r), Eqs. 16-18 result in an error term which depends upon the weighting scheme; for example the linear weighting function results in a error proportional to Δ_r^2.

This can be implemented in a standard linear weighting scheme simply by modifying the volumes used to compute n_j. Taking the cell size in the z-direction to be Δ_z, the volumes become:

$$V_j = \Delta_z \frac{\pi}{3}\left[r_{j+1}\left(r_j + r_{j+1}\right) - r_{j-1}\left(r_{j-1} + r_j\right)\right], \tag{19}$$

$$V_0 = \Delta_z \frac{\pi}{3}\left(r_1 - r_0\right)\left(2r_0 + r_1\right), \text{ and} \tag{20}$$

$$V_N = \Delta_z \frac{\pi}{3}\left(r_N - r_{N-1}\right)\left(r_{N-1} + 2r_N\right). \tag{21}$$

The corrected and uncorrected densities have been implemented in the XOOPIC [4] and XPDC1 [5] codes, and comparisons of the results are shown in Fig. 1. The comparisons shown are valid when there is at least 1 particle per cell. Total charge and volume are also identically conserved with the corrected method. This method can be easily extended to arbitrary weighting functions by using the desired weighting in Eqs. 16-18.

For linear weighting in spherical coordinates the density can be written:

$$n_j = \frac{\int_{r_{j-1}}^{r_j} f(r)\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} f(r)\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}{\int_{r_{j-1}}^{r_j} 4\pi r^2\frac{r-r_{j-1}}{r_j-r_{j-1}}dr + \int_{r_j}^{r_{j+1}} 4\pi r^2\frac{r_{j+1}-r}{r_{j+1}-r_j}dr}, \tag{22}$$

with edge densities again obtained by dropping out of bounds integrals.

For the uniform particle distribution, $f(r) = 4\pi r^2$, we again obtain the exact solution for all $0 \leq j \leq N$, $n_j = 1$. For more general particle distributions we obtain the same result as that of the cylindrical case above.

Similar to the cylindrical case, the method can be implemented for the spherical scheme by precomputing the volumes using the denominator of Eq. 22:

$$V_j = \frac{\pi}{3}\left(r_{j+1} - r_{j-1}\right)\left(r_{j-1}^2 + r_{j-1}r_j + r_{j-1}r_{j+1} + r_j^2 + r_j r_{j+1} + r_{j+1}^2\right), \quad (23)$$

$$V_0 = \frac{\pi}{3}\left(r_1 - r_0\right)\left(3r_0^2 + 2r_1 r_0 + r_1^2\right), \text{ and} \quad (24)$$

$$V_N = \frac{\pi}{3}\left(r_N - r_{N-1}\right)\left(r_{N-1}^2 + 2r_{N-1}r_N + 3r_N^2\right). \quad (25)$$

The densities computed with the method presented here are compared to the uncorrected densities for linear weighting on a uniform and nonuniform mesh in Fig. 2.

4. CURRENT DENSITY

This algorithm can be applied to current density for the electromagnetic source term in a straightforward manner by using $J = nqv$. The quantity qv is weighted to the mesh for each particle, and the result is divided by the surface area, **S**. To obtain the surface area, consider the differential elements of the components in cylindrical coordinates,

$$dS_r = rd\theta dz, \quad (26)$$

$$dS_\theta = drdz, \text{ and} \quad (27)$$

$$dS_z = rdrd\theta. \quad (28)$$

It is evident from Eqs. 26-28 that only the axial component of **S** will result in a nonlinear dependence on r when computing the surface area from $S = \int W dS$, where W is the particle-mesh weighting used. The results for the axial component of the current density are trivially different from the charge density results derived above, and the comparison plots are identical.

5. CONCLUSIONS

An algorithm for obtaining the correct charge and current densities in curvilinear coordinate systems for arbitrary particle interpolation schemes is described. Volumes and surface areas are weighted to the mesh using the same interpolation scheme used to weight particles. The method recovers the charge density correction factors for linear weighting in cylindrical coordinates [2], but extends to the more general case. The method has the notable beneficial properties of conserving charge and current as well as total volume and surface area on a general orthogonal mesh. The algorithm yields the exact answer for the uniform particle distribution, as demonstrated here. Arbitrary distributions are similarly correct to the mesh resolution, becoming exact as the mesh size approaches zero for continuum distributions; the corresponding electric field can also be shown to approach the exact solution in the same limit. This is a significant conceptual departure of the previous scheme of computing using incorrect volumes and then using correction factors designed for special cases [2]. The results are easily extended to two and three dimensions.

ACKNOWLEDGMENT

This work supported in part by the Air Force Office of Scientific Research STTR contract F49620-99-C-0028, Department of Energy SBIR contract DE-FG03-99ER82903, and General Electric Corporation contract GE-20000181-BIRDSALL-02/02. The author is grateful for useful discussions with D. L. Bruhwiler, H. B. Smith, H. J. Lee, K. L. Cartwright, A. B. Langdon, and C. K. Birdsall.

REFERENCES

1. W. M. Ruyten. Density-conserving shape factors for particle simulation in cylindrical and spherical coordinates. *Journal of Computational Physics*, 105:224–232, 1993.

2. D. J. Larson, D. W. Hewett, and A. B. Langdon. Correction factors for PIC accumulation on radial grids. *Computer Physics Communications*, 90:260–266, 1995.

3. C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* McGraw-Hill, New York N.Y., 1985.

4. J. P. Verboncoeur, A. B. Langdon, and N. T. Gladd. An object-oriented electromagnetic PIC code. *Comp. Phys. Comm.*, 87:199–211, 1995.

5. J. P. Verboncoeur, M. V. Alves, V. Vahedi, and C. K. Birdsall. Simultaneous potential and circuit solution for 1d bounded plasma particle simulation codes. *J. Comp. Phys.*, 104:321–328, 1993.

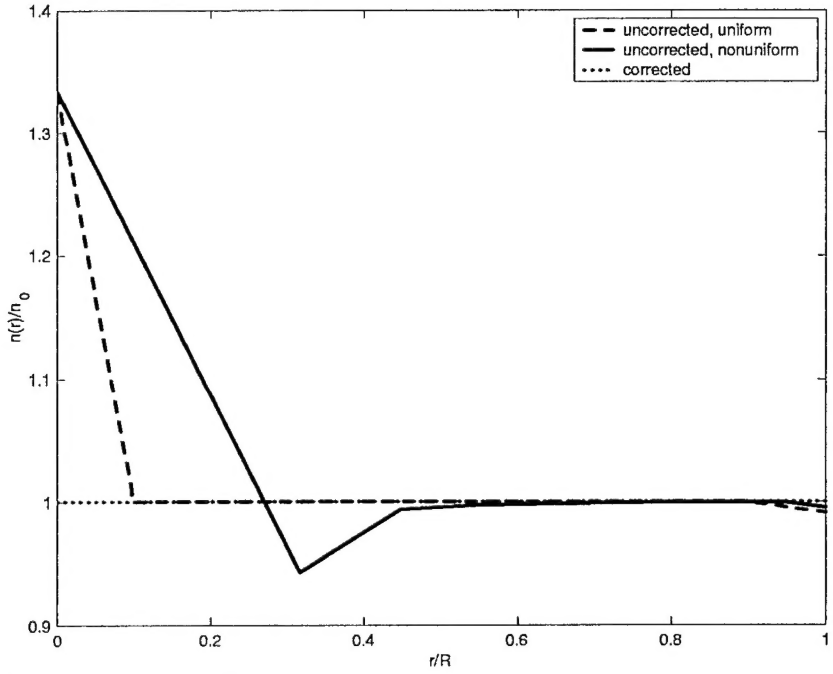


FIG. 1. Mesh densities computed using standard linear weighting (uncorrected) on a uniform and nonuniform mesh in cylindrical coordinates, as well as the corrected density on both meshes. The uniform mesh has $N = 10$ steps; the nonuniform mesh is computed from $r_j = \sqrt{r_{j-1}^2 + \left(r_N^2 - r_0^2\right)/N}$, such that the interval contains $N = 10$ equal volumes.

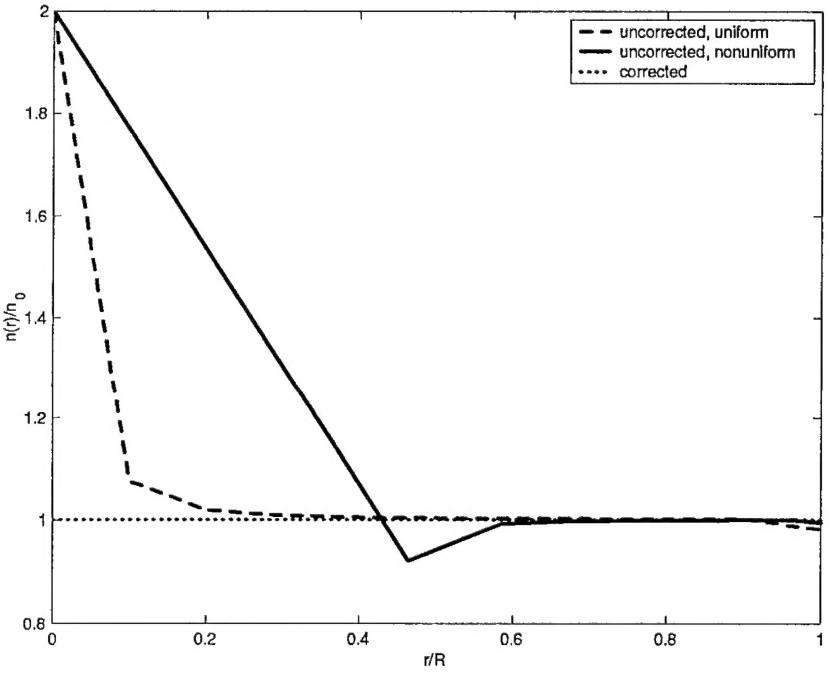


FIG. 2. Mesh densities computed using standard linear weighting (uncorrected) on a uniform and nonuniform mesh in spherical coordinates, as well as the corrected density on both meshes. The uniform mesh has $N = 10$ steps; the nonuniform mesh is computed from $r_j = \left[r_{j-1}^3 + \left(r_N^3 - r_0^3\right)/N\right]^{1/3}$, such that the interval contains $N = 10$ equal volumes.